AD A138048

EFFECT OF DATA LATENCY
UPON MISSILE ACCURACY

THESIS

Lee J. Monroe
Captain USAF

AFIT/GE/EE/83D-49

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY (ATC)
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

84 02 17 009

AFIT/GE/EE/83D-49

Accession For		
NTIS GRA&I	☒	
DTIC TAB	☐	
Unannounced	☐	
Justification		
By		
Distribution/		
Availability Codes		
Dist	Avail and/or Special	
A-1		

EFFECT OF DATA LATENCY
UPON MISSILE ACCURACY

THESIS

Lee J. Monroe
Captain USAF

AFIT/GE/EE/83D-49

AFIT/GE/EE/83D-49

EFFECT OF DATA LATENCY
UPON MISSILE ACCURACY

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

by

Lee J. Monroe, B.S.

Captain USAF

Graduate Electrical Engineering

December 1983

Preface

The purpose of this study is to examine the effects of data latency upon missile accuracy. This project was sponsored by the Midcourse Guidance Section (DLMM) of the Air Force Armament Laboratory, Eglin Air Force Base, Florida.

This study could not have been performed without the assistance of Mr. Marlow Henne, AFATL/DLMM; Dr. Gary Lamont, AFIT/EN; Lt. Col. Bob Edwards, AFIT/EN; and 1Lt. Geoff Donatelli, AFWAL/FIA.

In addition, I wish to thank the Lord for the strength to see this work through. To Him be the glory!

Table of Contents

Table of Contents (Cont'd)

List of Figures

List of Tables

List of Symbols

Acom	Commanded Acceleration
$Acom_D$	Commanded Acceleration (from the last DT cycle)
Aout	Output Acceleration
Apdlay	Total delays preceding the autopilot loop
A_3	Target achieved lateral acceleration
c	Speed of light
$(C_{DO})_{BOOST}$	Axial zero lift drag coefficient during boost
$C_{M\sigma}$	Slope of pitching moment coefficient versus deflection angle
$C_{N\alpha}$	Slope of normal force coefficient versus angle of attack
Delay1	Digital delays between the seeker and the guidance computer
Delay2	Delays between the guidance computer and the autopilot loop
Delay3	Delays between the autopilot loop and the guidance computer
DIS	Digital Integrating Subsystem
DISMUX	Digital Integrating Subsystem Multiplex Bus
DOF	Degrees of Freedom
DT	Basic integration interval in Tactics IV
g	Acceleration due to gravity
K	10^3
M	Mach Number
PRF	Pulse repetition frequency
Str/Lvl	Straight and level
Tcomp	Time at which guidance computer forms Acom

List of Symbols (Cont'd)

T_c	Autopilot time constant
T_{cl}	Lumped time constant
$T_{cl(min)}$	Minimum lumped time constant
$T_{cl(max)}$	Maximum lumped time constant
t_s	Settling time
$t_{s(min)}$	Minimum settling time
$t_{s(max)}$	Maximum settling time
V_{23}	Missile - Target closure rate
V_2	Missile velocity
V_3	Target velocity
α	Angle of attack
$\dot{\gamma}_3$	Turn rate
σ	Deflection angle
$(\Delta C_{DO})_{COAST}$	Incremental zero lift drag coefficient increase during coast
ε	Antenna Boresight Error
ζ	Autopilot damping ratio
λ	Navigation constant
η	Seeker Azimuth angle
ξ	Seeker elevation angle
π	3.141592..
ω_{23}	Line of sight rate
ω_n	Autopilot undamped natural frequency
Z.O.H.	Zero Order Hold

Abstract

This study examined the effect of data latency upon air/air guided missile accuracy. This research was done by modeling a digital guided missile, inserting the model into a computer simulation and generating miss distance statistics. The digital guided missile was modeled after the DIS microcomputer architecture. The DIS (Digital Integrating Subsystem) approach involves a number of loosely coupled microprocessors which communicate over a serial multiplex bus. It was developed at the Air Force Armament Lab~~oratory~~, Eglin AFB, FL. The missile simulation ~~was~~ Tactics IV, ~~This simulation~~ involves three degrees of freedom and is written in Fortran IV. It was developed by Science Applications, Inc in conjunction with AFWAL/FIMB, Wright Patterson AFB, OH. The results of this study indicate that typical data latency values generate only small increases in miss distance. The maximum delays tested were .01 seconds and the average increase in miss distance was 2.12 feet. Additionally, it was discovered that the transmission rate of the DIS microcomputers greatly affected miss distance. Microcomputers transmitting at 10 HZ generated large miss distances, even without data latency present. The identical missile engagements using transmission rates of 100 HZ resulted in much smaller miss distances.

EFFECT OF DATA LATENCY
UPON MISSILE ACCURACY

I. Introduction

Background

The Air Force has developed a digital control system for guided missiles known as the Digital Integrating Subsystem (DIS) (Ref 7). With this system, the missile computational tasks are divided among a number of loosely coupled microcomputers. These microcomputers communicate over a serial, multiplex bus known as DISMUX. This approach boasts many advantages over traditional, custom-built missile control systems. One of these advantages is that the DIS computers are based upon inexpensive, general purpose microprocessors. The other is that the loosely coupled nature of DIS promotes interoperability between different missile systems. However, along with these benefits come disadvantages including "data latency".

"Data latency" describes the time delays in a digital guided missile. These delays include microcomputer bus delays and computation delays. These delays are termed a disadvantage in that they can clearly decrease stability (Ref 11). What is not clear, however, is how they effect missile accuracy. It is this question which this work addresses.

Problem Statement and Approach

This thesis related "data latency" to missile accuracy by first modeling a DIS missile. The model included a microcomputer bus network, a radar seeker, a guidance computer and an autopilot. The DIS time delays and errors were also included.

Next, the model was implemented with the Tactics IV computer simulation. The missile modules in the simulation were modified so as to reflect the DIS model. This modified program is named revised Tactics IV.

Then, a series of test cases were established which relate data latency to missile accuracy. The variables include the size of the digital delays, the target maneuver, the attack geometry and the transmission rates of the DIS computers. Results included miss distance statistics and plots of key engagement variables.

Next, a tutorial summary of the factors which affect miss distance was developed. This summary demonstrates that data latency is merely a part of one miss distance factor (missile response time). A grasp of all the factors is required to understand the test results.

Finally, the miss distance statistics were compiled into six tables. These tables indicate how miss distance varied with the amount of data latency. Additionally, the key engagement variable plots were examined to determine how the other miss distance factors affected the engagement.

Assumptions and Parameters

The major assumptions in this research were those inherent in the original Tactics IV simulation and those made in constructing revised Tactics IV. The original Tactics IV assumptions will be presented first.

The missile model in Tactics IV has 3 degrees of freedom. Therefore, the model did not include higher order motions such as bending. This reduced both the size of Tactics IV and processing time. The penalty was paid in terms of a less accurate missile model (Ref 6).

Secondly, missile aerodynamic response was treated as a simple second order system. The damping ratio and natural frequency of the response were considered "fixed". Missiles are more correctly expressed by higher order equations. In these higher order equations, the damping ratio and natural frequency are dynamic variables. The simple second order response was chosen for ease of implementation. Additionally, the constant natural frequency assumption was accurate if dynamic pressure was assumed constant (Ref 5).

The missile guidance law employed is a form of proportional navigation. It is assumed that the required lateral acceleration is directly proportional to the line of sight rate times missile velocity. This differs from the standard proportional navigation law. With the standard law, lateral acceleration is directly proportional to the line of sight rate times missile closing velocity. This choice of laws is practical as not all missiles possess equipment for measuring closing velocity. This law produced inflated accelerations at launch, as missile velocity was at a maximum then. At the end of the engagement, commanded accelerations were low as missile velocity had dropped (Ref 6).

Another assumption used in original Tactics IV is that the inertial anglular rotation of the earth is negligible. This assumption avoids rotation of the reference frames over time. This assumption is valid over the short flight times involved (Ref 6,9).

Also note that the seeker is colocated with the missile center of gravity. This assumption is reasonable as the distance from the seeker to the missile center of gravity is negligible compared to the range to the target. This assumption avoids an additional transformation from a seeker frame to the missile body frame (Ref 6,9).

Finally, all the missile and target turns were assumed coordinated. This assumption neglected the energy losses associated with uncoordinated flights. It also discounted the false line of sight rates that uncoordinated turns can indicate. Coordinated turns were chosen to simplify motion modeling (Ref 6,4).

The first assumption associated with the revised Tactics IV program is that all of the computers in the DIS model transmitted data at the same rate. This assumption is based upon data presented in a previous DIS data latency study (Ref 3). The value of this assumption is that the transmission interval could be set equal to the basic integration interval in Tactics IV. The transmission rates examined were 100 HZ and 10 HZ.

The second assumption was that all the computers attempted to transmit at the beginning of the integration interval. This assumption provided a reference frame for defining the various digital delays.

The third assumption associated with revised Tactics IV was that both analog-digital converter error and finite word length error was zero mean, normally distributed and uncorrelated to signal level. These assumptions are justified in the literature (Ref 7).

The fourth assumption associated with revised Tactics IV was that the total digital delays did not exceed .01 seconds. This was based on the data contained in earlier DIS studies. In these studies, delays were of the order of microseconds. (Ref 12).

The fifth assumption was that the seeker unit has a fixed lag of .1 seconds. This figure was based upon typical seeker figures (Ref 4). This approach was deemed superior to the original Tactics IV technique of employing a single guidance loop delay.

The final assumption used in revised Tactics IV was that the autopilot loop response was second order. As mentioned, this assumption is based on the fact that the missile aerodynamic response is much slower than that of the actuator loop (Ref 4). Necessarily, this assumption lumps all of the delays in the autopilot loop under the loop lag. This assumption also ignores error sources such as digital-analog converter error, inertial reference errors, ect. This simplified autopilot response was chosen for ease of analysis and implementation.

II. DIS Missile Model

Introduction

The DIS missile model is a series of diagrams and equations which mathematically describe the behavior of a DIS guided missile. This model was developed so as to explain how data latency enters the control loop. It was also designed with the knowledge that the model would be implemented with the Tactics IV computer simulation.

This chapter begins by modeling the complete DIS loop. This introduces the various loop components along with various delays and errors. Next, the engagement reference frames are presented. These frames are necessary to understand the detailed operation of the loop components. Finally, the loop components are thoroughly modeled.

DIS Control Loop

A guided missile control loop is a collection of components which sense a target and fly the missile to impact with the target. The DIS system consists of five major components which perform this task (see Fig. 1). The components are the seeker, the guidance computer, the digital autopilot, the actuator computer, and finally, the inertial reference unit (Ref 7). The task of each of these elements will now be presented.

The seeker senses the targets position and passes information about the target to the guidance computer. The guidance computer takes this target information and computes a commanded acceleration which should fly the missile to the target. The digital autopilot takes the acceleration commands and converts them to a commanded control surface deflection.

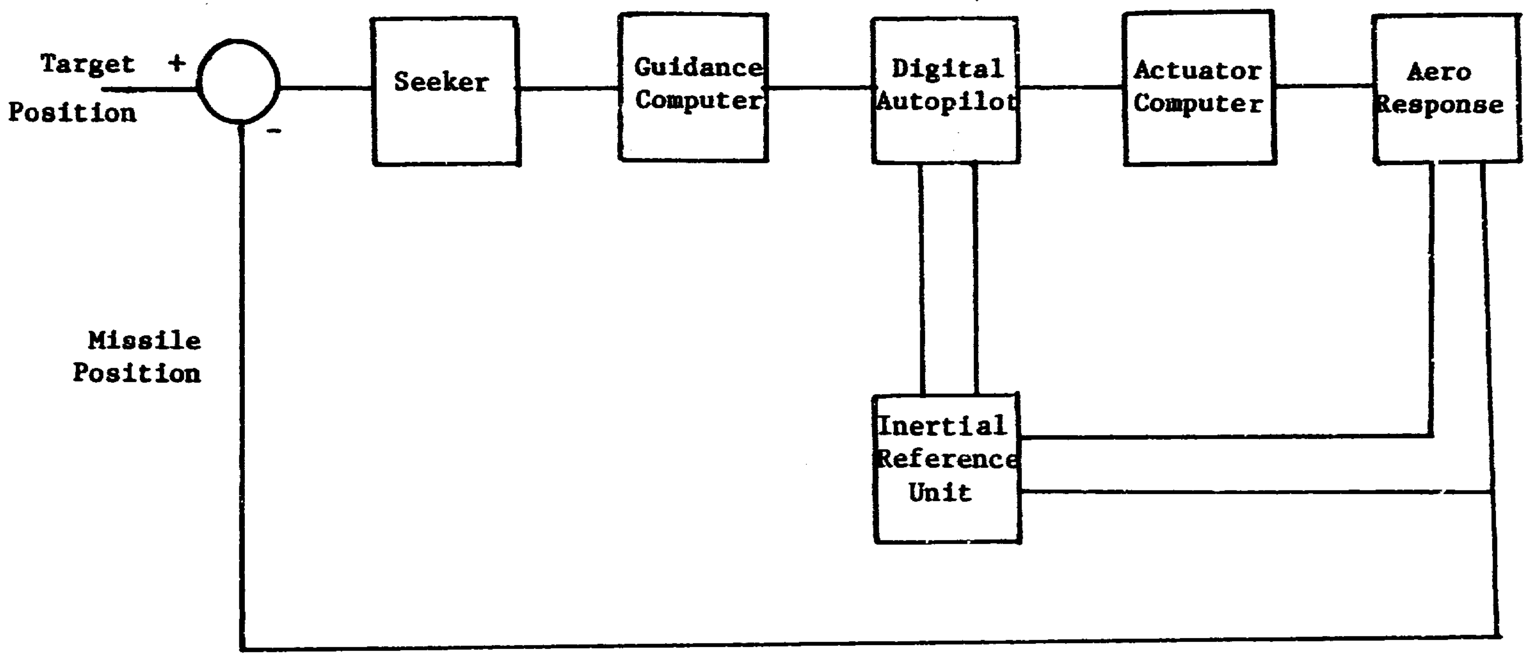


Figure 1. Basic DIS System

This deflection should produce the commanded acceleration. Next, the actuator computer takes the commanded control surface defelection and commands the control surfaces to that position. Once the control surfaces are moved, an aerodynamic response results. The inertial reference unit senses acceleration and angular rate and estimates velocity and acceleration. These data are then returned to the digital autopilot. This completes the overview of the basic control loop. All that remains is to insert the data latency characteristics of DIS.

The complete DIS system is shown in Fig. 2. The first additions to the basic loop are a sampler, a zero order hold and an analog-digital converter (Ref 11). These three components put the seeker signals into the proper digital form. As will be shown later, these components also inject a converter error and a delay for the conversion. Now the seeker data is ready for transmission to the guidance computer.

The next addition to the basic DIS system is that of bus delay. This delay occurs for two reasons. The first is that the seeker's bus control unit must "wait its turn" before it can transmit on the bus. The second is that the data transmission requires time (Ref 5). Once this bus delay elapses, the seeker data is available for use by the guidance computer.

The guidance computer calculates commanded accelerations. It does this according to a guidance law known as "proportional navigation". The computation injects some error due to the finite word length of the computer. Also, a computation time delay is introduced. Now, the guidance computer output data is ready for transmission on the DISMUX bus.

The digital autopilot accepts digital data from the guidance computer and the inertial reference unit. It does this after the data incurs a

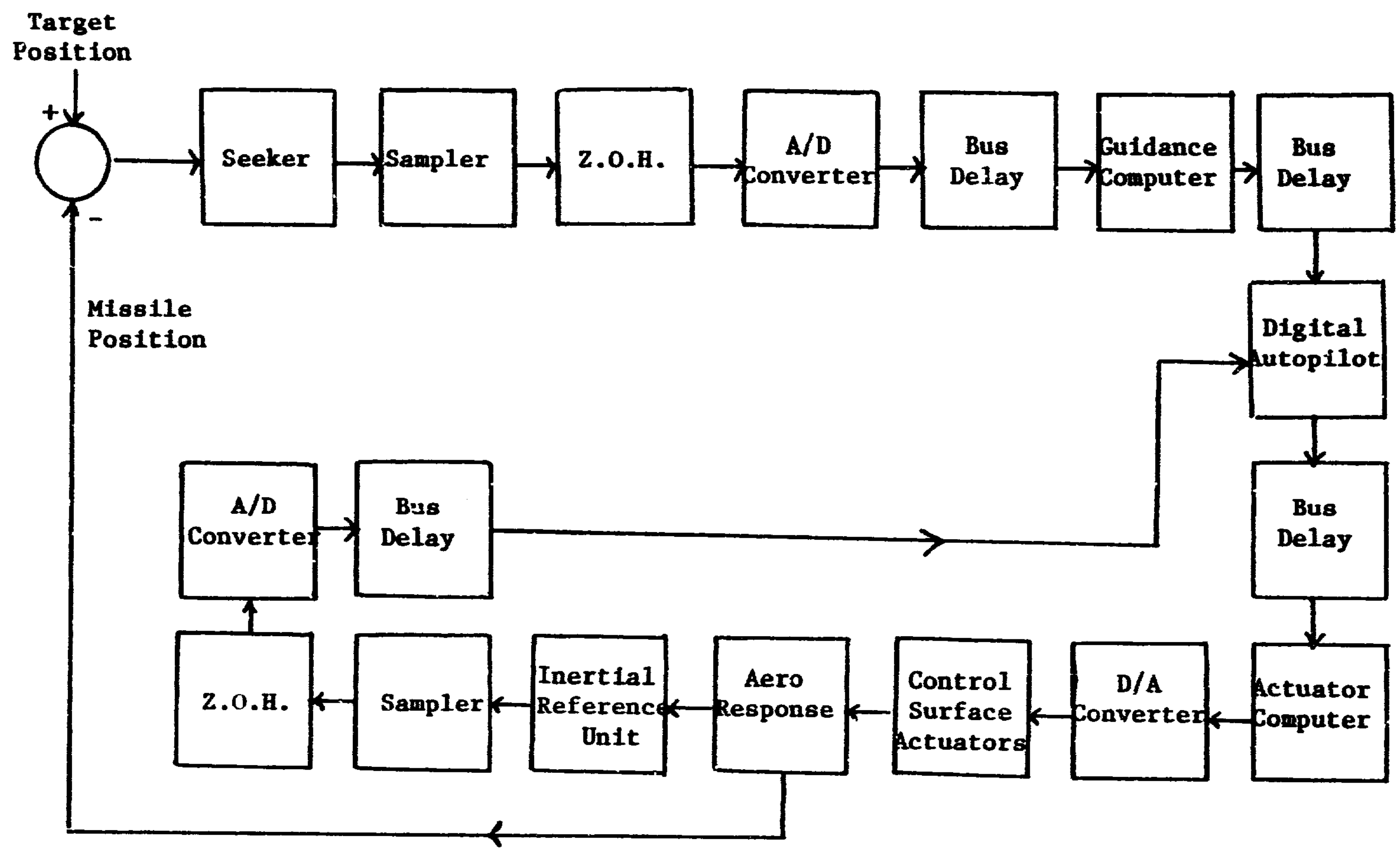


Figure 2. Complete DIS System

bus delay. It then calculates the commanded fin deflection while introducing a computation error. Also, a computation time delay is introduced. The autopilot output is then put on the bus.

The actuator computer receives the commanded fin deflection from the bus. It then forwards this signal to a digital-analog converter. This device converts the digital data to analog form. This process injects a converter error and time delay. Finally, the servo moves the control surface.

The inertial reference unit is the final element in the control loop. The inertial sensors produce analog estimates of velocity and acceleration. These measurements are sampled, held and passed through an analog-digital converter. Again, the converter injects a conversion error and a time lag. Finally, the digital output is put on the bus for transmission to the digital autopilot. With the DIS control loop defined, the reference frames for the model will now be presented.

Reference Frames

In order to define the missile engagement, it was first necessary to develop reference frames. These frames describe the position and translation of the missile and target through simulation space. The two reference frames are inertial reference and missile body frames. Additionally, two important seeker angles were required (Ref 6).

The inertial frame is centered at mean sea level on a flat earth below the missile point mass. The three orthogonal axes are x, y and z. These axes are constrained so that the xy axes lie in the flat earth plane and the z axis is orthogonal to the plane.

Four related notations are developed within the inertial frame.

These notations are absolute and relative position and absolute and relative velocity. This development is somewhat cumbersome but was adapted so as to remain consistent with the Tactics IV simulation.

Absolute position is used to describe the setup of a missile/target engagement. It consists of both Cartesian and spherical coordinates. The Cartesian coordinates are (x_i, y_i, z_i) as depicted in Fig. 3. As mentioned, z_i is in feet above Mean Sea Level. The spherical coordinates are $(r_i, \theta_i, \gamma_i)$, also as depicted in Fig. 3. An additional note is the use of the subscript "i". It refers to the three engagement bodies as given in Table I.

Relative position is used to describe the missile intercept problem. It consists of both Cartesian and spherical coordinates. The Cartesian coordinates are (x_{ij}, y_{ij}, z_{ij}) and the spherical coordinates are $(r_{ij}, \theta_{ij}, \gamma_{ij})$. This notation is also depicted in Fig. 3. An important note is that the subscript "ij" refers to the directed distance from "i to j". Thus in Fig. 3, r_{12} refers to the directed distance from 1 to 2.

Absolute velocity is used to describe the setup of the engagement. It consists of both Cartesian and spherical coordinates. The Cartesian coordinates are (V_{xi}, V_{yi}, V_{zi}) and the spherical coordinates are $(V_i, \theta_{vi}, \gamma_{vi})$. This notation is depicted in Fig. 4.

Relative velocity is used to describe the missile intercept problem. It consists of Cartesian and spherical coordinates. The Cartesian coordinates are $(v_{xij}, v_{yij}, v_{zij})$ and the spherical coordinates are $(V_{ij}, \theta_{vij}, \gamma_{vij})$. Additionally, the relative velocity notation includes two other constructs. These constructs are the "relative range rate vector" $(\dot{\bar{r}}_{ij})$ and the "angular rate vector" $(\bar{\omega}_{ij})$. The relative range rate vector

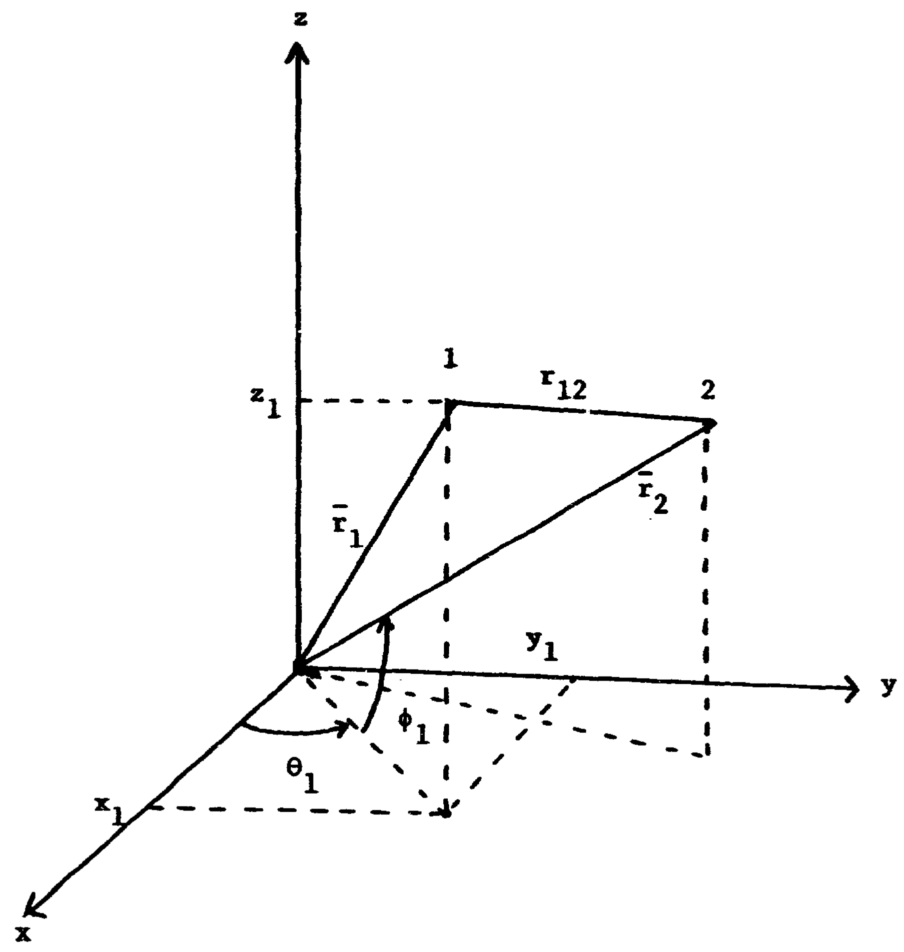


Figure 3. Inertial Position

Table I

Body Subscripts

i	Body
1	Launch Aircraft
2	Missile
3	Target Aircraft

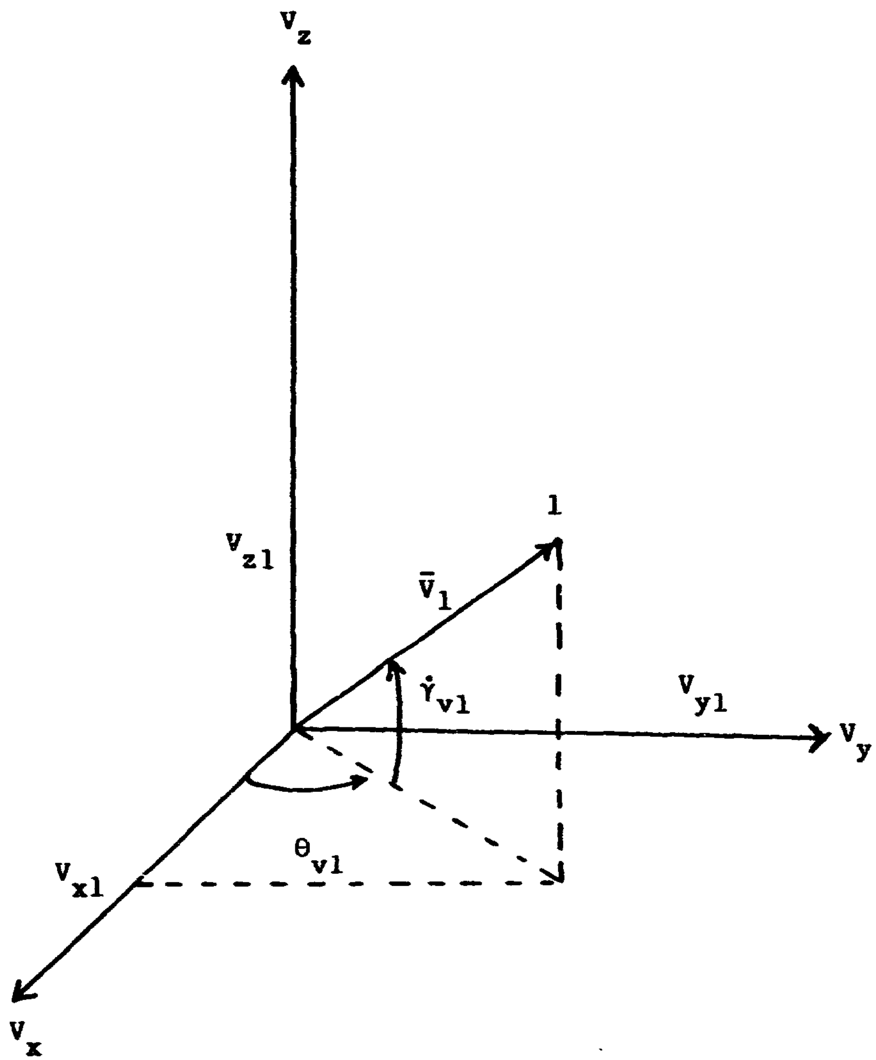


Figure 4. Inertial Velocity

is the time rate of change of the relative range vector $\bar{r}_{ij}$, presented earlier. This construct is important in the guidance computation. It is defined by the following equation:

$$\dot{\bar{r}}_{ij} = \dot{r}_{ij}\, \bar{1}_{rij} + \bar{\omega}_{ij} \times \bar{r}_{ij} \tag{1}$$

where

$\dot{r}_{ij}$ = scalar time rate of change of $\bar{r}_{ij}$

$\bar{1}_{rij}$ = unit vector along $\bar{r}_{ij}$

$\bar{\omega}_{ij}$ = angular rate vector orthogonal to both $\bar{r}_{ij}$ and $\dot{\bar{r}}_{ij}$

$\bar{r}_{ij}$ = relative range vector between i and j

The second construct, the angular rate vector, is essential in the missile guidance computation. It also known as the line of sight rate (LOS) and is defined by the following:

$$\bar{\omega}_{ij} = \bar{r}_{ij} \times \left(\frac{\bar{v}_j - \bar{v}_i}{r_{ij}^2}\right) \tag{2}$$

where

$\bar{r}_{ij}$ = relative range vector between i and j

$\bar{v}_j$ = velocity vector of body j

$\bar{v}_i$ = velocity vector of body i

r_{ij} = scalar relative range between body i and j

This formula completes the inertial reference frame.

The missile body frame describes the orientation of the missile in inertial space. The missile is symmetrical and cruciform as depicted in Fig. 5. The three orthogonal axes are pitch ($\bar{1}_p$), roll ($\bar{1}_r$), and yaw ($\bar{1}_y$) as shown in Fig. 5.

The two Seeker angles relate the target relative range vector to the missile body frame. These two angles are the azimuth angle (η) and the elevation angle (ξ). Together they define the "line of sight" angle to the apparent radar target. The rate of change of this angle, ω_{ij}, was presented in the velocity frame. It is this variable which is used by the guidance computer to produce commanded acceleration. The two seeker angles are presented in Fig. 6 and are defined by:

$$\eta = \tan^{-1} \left(\frac{\bar{1}_{r12} \cdot \bar{1}_p}{\bar{1}_{r12} \cdot \bar{1}_y}\right) \tag{3}$$

$$\xi = \sin^{-1} (\bar{1}_{r12} \cdot \bar{1}_y) \tag{4}$$

where

$$\bar{1}_{r12} = \frac{\bar{r}_{12}}{r_{12}}$$

With the two seeker angles thus presented, the necessary references have been defined for the intercept.

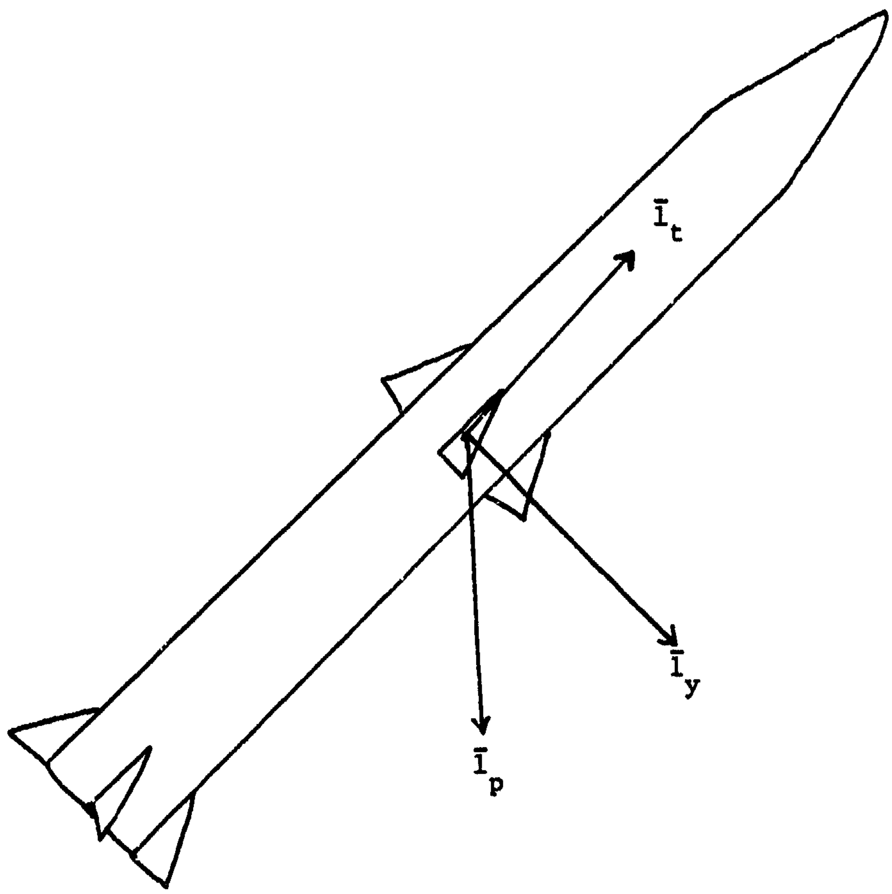


Figure 5. Cruciform Missile

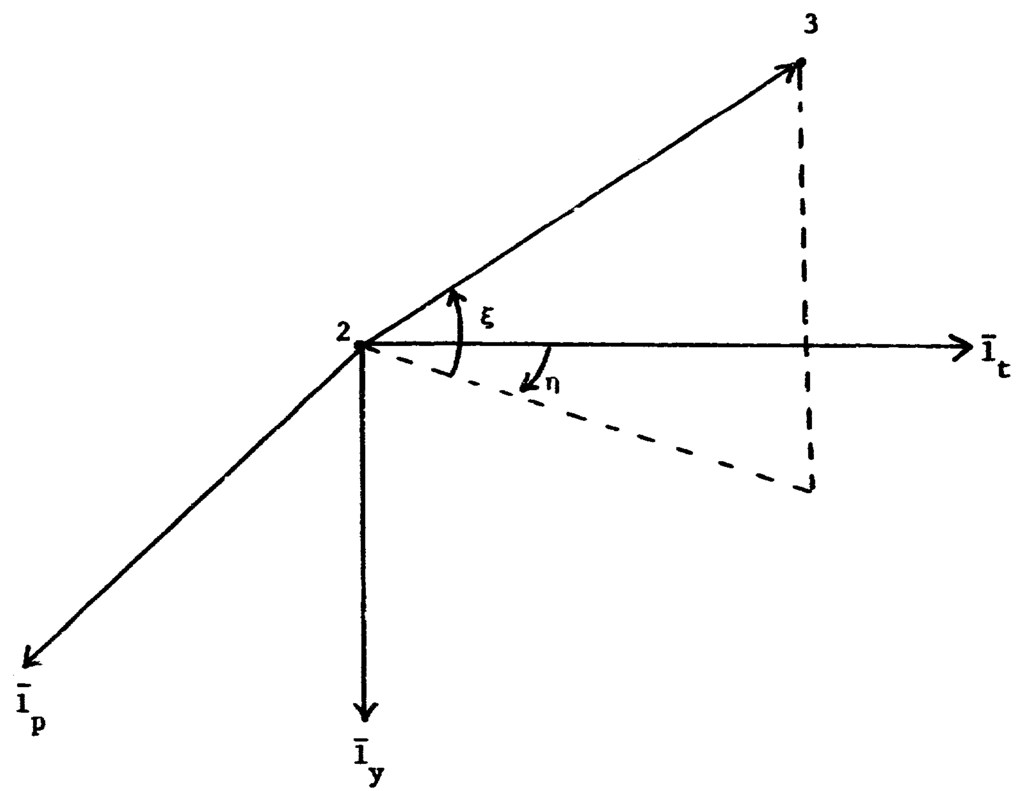


Figure 6. Seeker Angles

Seeker Model

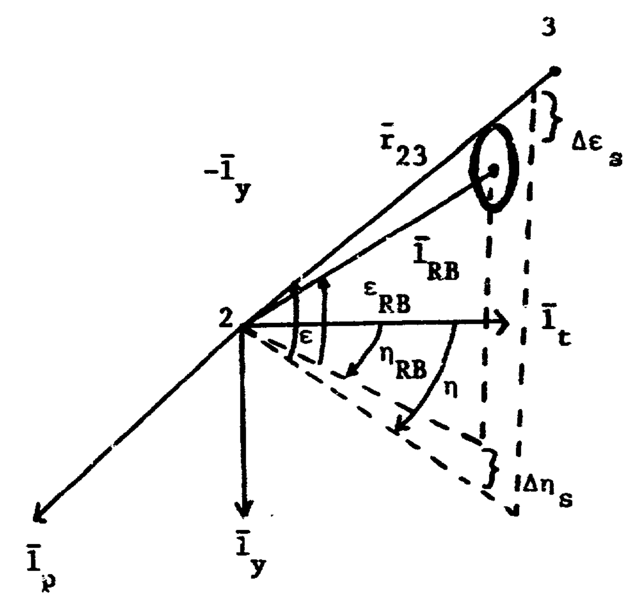
The first component in the DIS control loop is the seeker. The function of the seeker is to sense the apparent radar target location and to pass the target line of sight rate to the guidance computer. The three components which perform this function are the radar sensor, the drivers and the driver rate sensors (Ref 9). These components will be discussed and then the seeker noises and errors introduced.

The radar sensor detects the apparent target radar centroid. It outputs a signal proportional to the angle between this apparent centroid and where the radar is currently pointing (radar boresight). To understand this operation, the idea of radar boresight will be developed along with sensor error angles.

The radar boresight is where the radar is currently pointing. The radar boresight is normally expressed as a unit vector, $\bar{I}_{RB}$, and its orientation is determined by the boresight azimuth angle (   ) and boresight elevation angle (ω_{RB}) (See Fig. 7). These two angles are analogous to the azimuth and elevation angles introduced earlier. The difference is that they relate the radar to the missile body frame instead of the apparent radar target to the missile body frame.

The sensor error angle is the angular difference between the radar boresight and the apparent radar centroid. The two components of this angle are the azimuth error angle and the elevation error angle. They are depicted in Fig. 7 and are defined as:

$$\Delta\eta_s = \eta - \eta_{RB} \tag{6}$$



$\Delta\eta s$: Azimuth Error Angle

$\Delta\epsilon s$: Elevation Error Angle

Figure 7. Sensor Error Angles

$$\Delta\xi_{s} = \xi - \xi_{RB} \tag{7}$$

As mentioned, the sensor passes a signal proportional to these error angles to the drivers.

The drivers strive to null the seeker error angles. The mechanism employed is depicted in Fig. 8. Notice that the seeker drivers are first order lag networks whose outputs are azimuth rate ($\dot{\eta}_s$) and the seeker elevation rate ($\dot{\xi}_s$). The transfer functions which relate these outputs to the error angle inputs are:

For the azimuth driver:

$$\frac{\dot{\eta}_s}{\Delta\eta_s} = \left(\frac{Ks}{s + \frac{1}{T_1}}\right) \tag{7}$$

For the elevation driver:

$$\frac{\dot{\xi}_s}{\Delta\xi_s} = \left(\frac{Ks}{s + \frac{1}{T_1}}\right) \tag{8}$$

where

K = open loop gain

s = Laplace operator

T_1 = first order lag time constant

The final seeker component is the driver rate sensor. This device merely measures the seeker azimuth rate ($\dot{\eta}_s$) and elevation rate ($\dot{\xi}_s$) and forwards these signals to the guidance computer. With the seeker

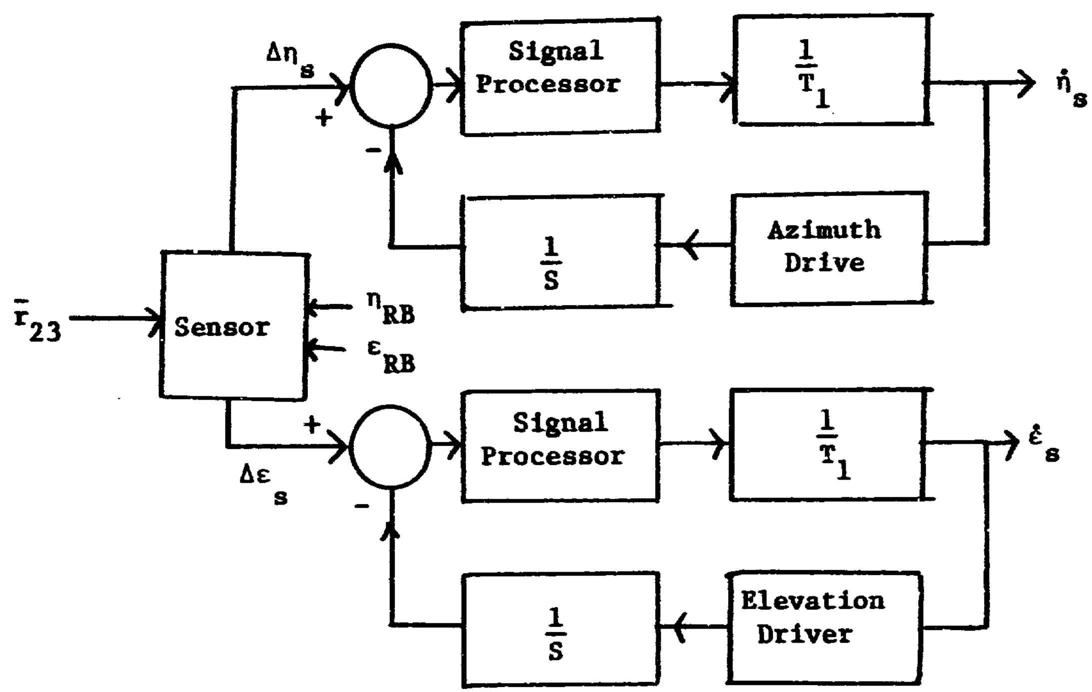


Figure 8. Seeker Drivers

components explained, all that remain is to discuss the seeker noises and errors which were modeled (See Fig. 9).

The first seeker error is the antenna boresight error (ε). This error results when the received radar signal passes through the missile radome. When this occurs, the true line of sight is corrupted (Ref 6). This corruption effects the components of the line of sight, azimuth (η) and elevation (ε). The azimuth error is called ε_1 and the elevation error is called ε_2. The formulas used to compute these errors are:

$$\varepsilon_1 = \frac{R}{12} e^{-\eta/.525} \sin (2\phi) \sin (12\eta) \tag{9}$$

$$\varepsilon_2 = \frac{R}{12} e^{-\xi/.525} \sin (2\phi) \sin (12\xi) \tag{10}$$

where

η = azimuth angle

ξ = elevation angle

ϕ = seeker roll angle with respect to the body

R = boresight error slope

The boresight error slope (R) is calculated according to the following formula:

$$R = 0.05 \frac{\lambda}{d_s} (f_n - \tfrac{1}{2}) (1 + \frac{15\Delta f}{f}) \tag{11}$$

where

λ = radar wavelength

d_s = seeker diameter

F_n = nose finess ratio

$\frac{\Delta f}{f}$

These errors corrupt the azimuth and elevation angles according to these relationships:

$$\eta' = \eta + \varepsilon_1 \tag{12}$$

$$\xi' = \xi + \varepsilon_2 \tag{13}$$

These corrupted angles are then acted upon by the first seeker noise, thermal noise.

Thermal noise is a broad band noise which results from the random excitation of electrons in the seeker circuitry (Ref 6). Thermal noise is modeled as a white Gaussian noise directly proportional to the square of the distance to the target. This implies that (η') and (ξ') are corrupted according to the following formula (Ref 6):

$$\eta'' = \eta' + (n_{T1}\ r_{23}^{\ 2}) \tag{14}$$

$$\xi'' = \xi' + (n_{T2}\ r_{23}^{\ 2}) \tag{15}$$

where

n_{T1} = white Gaussian noise

n_{T2} = white Gaussian noise

r_{23} = distance from missile to target

Glint is the second seeker noise and it results from apparent target movement due to phasing errors. Glint is typically modeled as the output of a first order lag driven by white Gaussian noise. It is inversely proportional to the distance to the target. The two glint formulas are (Ref 6):

$$\eta''' = \eta'' + \left(\frac{n_{G1}}{r_{23}}\right) \tag{16}$$

$$\xi''' = \xi'' + \left(\frac{n_{G2}}{r_{23}}\right) \tag{17}$$

where

n_{G1} = the output of a first order lag driven by white Gaussian noise

n_{G2} = the output of a first order lag driven by white Gaussian noise

r_{23} = distance from missile to target

These two noise corrupted angles, η''' and ξ''', are then processed by the sensor as shown in Fig. 9.

The final seeker noise is injected into the seeker azimuth rate and seeker elevation rates during the Analog-Digital conversion process. As noted in Fig 9, these two signals are first sampled, then held and finally passed through the analog-digital converter. This final step introduces both a converter time delay and injects a converter noise. The noise

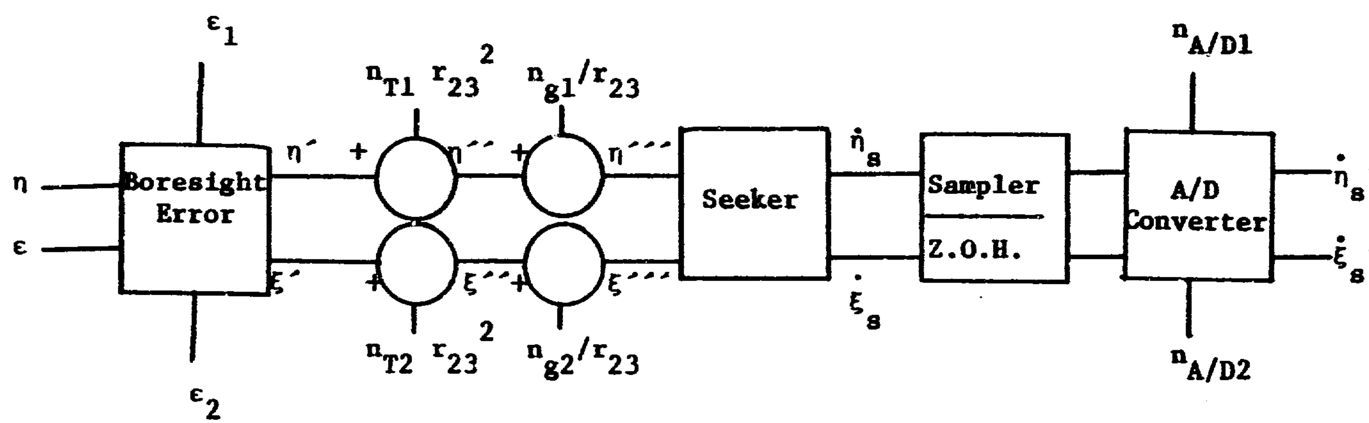


Figure 9. Seeker Noises and Errors

results because the incoming sampled analog signal never lies precisely at the digital signal level used to express it. The size of the error depends both on the number of digital bits of the converter and upon the analog voltage range the converter is designed to encounter. These two factors determine the number of "quantization" levels of the converter. The maximum error is ±½ the quantization level of the converter. The error actually is best expressed as a zero mean, normal distribution (Ref 3) whose standard deviation is:

$$\sigma_{AD} = \frac{q_{AD}}{12} \tag{18}$$

and the quantization level q is defined as:

$$q_{AD} = \frac{M}{2N_{AD}} \tag{19}$$

where

M = Maximum analog signal range

N_{AD} = Number of bits of the converter

With this final noise injected, the two digital seeker rates, $\dot{\eta}_s{}'$ and $\dot{\xi}_s{}'$, are forwarded to the guidance computer for further processing.

Guidance Computer

The next major component in the DIS loop is the guidance computer. The purpose of this computer is to issue the appropriate commands so as to drive the missile to the target. This is done by implementing a guidance law with the guidance computer. The guidance law employed is proportional navigation. This section will both develop the law and show how it was implemented. Additionally, the delays and errors associated with the guidance computer will be developed.

The proportional navigation guidance law is based on the observation that when the line of sight rate of the target is zero, the missile will impact the target. This phenomena is well known to aviators in that if an aircraft maintains a fixed angular position relative to your canopy, then a collision will eventually occur. The formula which expresses proportional navigation is as follows (Ref 4):

$$Acom = \lambda \ (\omega_{23} \times V_{23}) \tag{20}$$

where

Acom = missile acceleration required to null the line of sight rate

λ = navigation constant

ω_{23} = line of sight rate (magnitude)

V_{23} = closing velocity (magnitude)

With the guidance law introduced, each of the components of the law will be examined separately.

The missile acceleration required to null the line of sight rate, Acom, can be decomposed into two components. This first component is Acoma and is the acceleration required to null the line of sight rate in the horizontal plane. The second component is Acomd and is the acceleration required to null the line of sight rate in the vertical plane (Ref 6).

The navigation constant, λ, dictates the type of intercept the missile will fly. For $\lambda = 3$, the course will be a constant turn and will result in a "pursuit" intercept. For $\lambda = 5$, the intercept will begin with a hard turn and the missile will lead the target (Ref 4). The intermediate value of 4 is chosen.

The line of sight rate, ω_{23}, is composed of the azimuth rate ($\dot{\eta}$) and elevation rate ($\dot{\xi}$). As mentioned in the seeker block, the seeker provides $\dot{\eta}_s'$ and $\dot{\xi}_s'$ as the closest estimate of the desired rates.

The final variable, V_{23}, is the closing velocity of the missile. The missile must either sense this velocity or estimate it. This study approximates V_{23} by using the missile velocity V_2 instead. It is assumed that the inertial unit provides a perfect estimate of this velocity.

Combining this information yields the following two equations:

$$\text{Acoma} = (\lambda)\ (\dot{\eta}_s')\ (V_2) \tag{21}$$

$$\text{Acomd} = (\lambda)\ (\dot{\xi}_s')\ (V_2) \tag{22}$$

where

Acoma = horizontal component of lateral acceleration

Acomd = vertical component of lateral acceleration

λ = navigation constant

$\dot{\eta}_s'$ = seeker estimate of azimuth rate

$\dot{\xi}_s'$ = seeker estimate of elevation rate

V_2 = missile velocity

Notice that equation (22) is only accurate when the seeker azimuth rate is in the horizontal plane. Notice that equation (23) is only accurate when the seeker elevation rate is in the vertical plane. With these two assumptions in mind, the guidance law implementation is complete.

The first delay associated with the guidance computer is bus delay. The bus delay involves both the seeker data and the missile velocity data. The seeker data, $\ddot{\eta}_s'$ and $\dot{\xi}_s'$ are sent from the seeker unit over the DISMUX bus. Enroute, the data experiences a bus delay. The missile velocity is sent from the inertial reference unit and it also experiences a bus delay. These delays are shown in Fig. 10.

The second delay associated with the guidance computer is computation delay. This delay is the time required for the guidance computer to perform the guidance law calculations. This delay is shown as "compute delay" in Fig. 10.

The one error associated with the guidance computer is "finite word length" error. This error is the round off which occurs when binary numbers are multiplied together in the computer. This error is assumed to be zero mean and normally distributed (Ref 3). The standard deviation of the error (σ_{fwl}) is determined by the formula:

$$\sigma_{fwl} = \frac{q_{fwl}}{12} \tag{23}$$

where

q_{fwl} = quantization level

The quantization level (q_{fwl}) is determined by the number of bits in the digital computer. The formula is:

$$q_{fwl} = \frac{1}{2N_{fwl}} \tag{24}$$

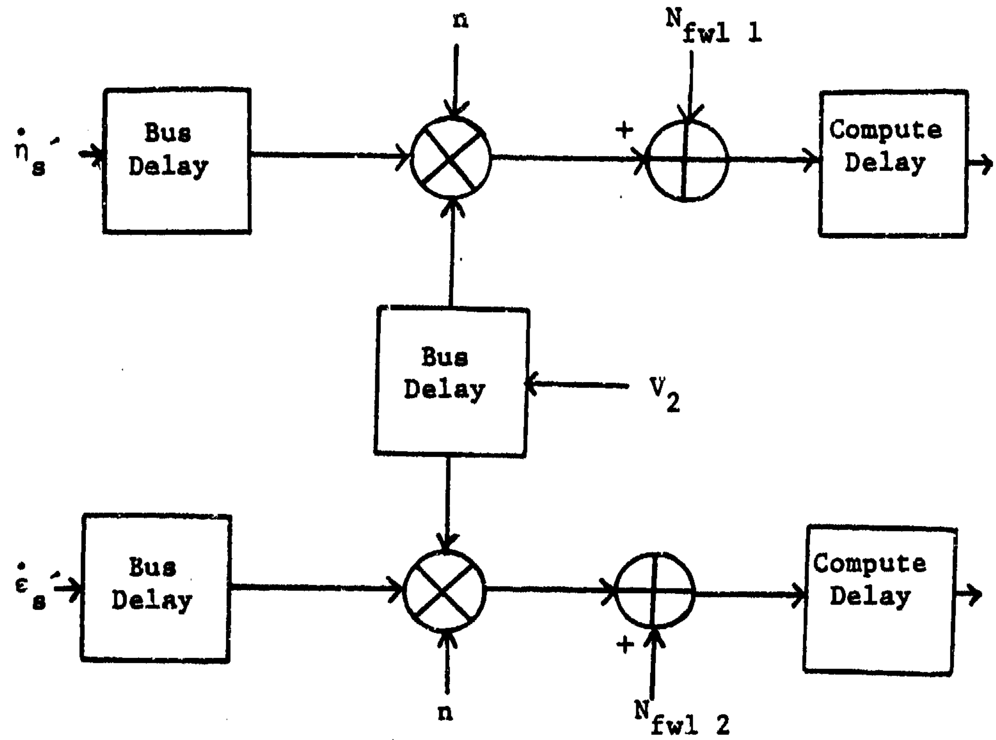


Figure 10. Guidance Computer

where

N_{fwl} = Number of bits in the computer

With this formula, the development of the guidance computer model is complete.

Autopilot Loop

The final element in the DIS loop is the autopilot loop. The purpose of this loop is to convert the accelerations commanded by the guidance computer into output accelerations. There are four components which form the loop and perform this function. These components are the digital autopilot computer, the control surface actuator computer, the control surface actuator and the inertial reference unit. These components will be explained separately along with the simplified model used to express the autopilot response.

The digital autopilot converts commanded accelerations into commanded control surface deflections. These deflections are computed based upon aerodynamic gains and inertial estimated velocity and acceleration. The commanded control surface defelections are then forwarded to the actuator computer (See Fig. 2).

The control surface actuator computer receives the commanded deflections from the DISMUX bus. The actuator computer merely forwards these data items to a digital-analog converter. This converter changes the signal to the required analog form.

The control surface actuator converts the commanded control surface deflections to actual deflections. The network which models this function is the simple first order lag system as shown in Fig. 11 (Ref 4).

The fourth component of the loop is the inertial reference unit.

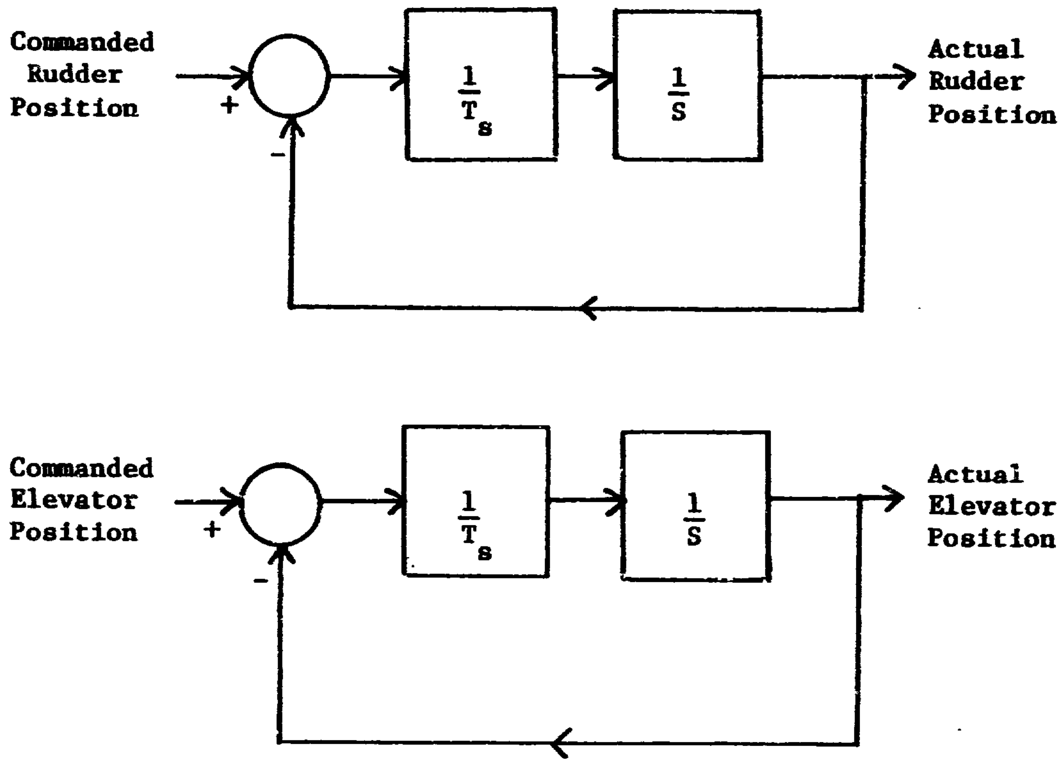


Figure 11. Control Surface Actuator Loop

This component has outputs of estimated velocity and acceleration. These analog signals are then sampled, held and converted to digital form (See Fig. 2). They then are sent over the DISMUX bus back to the digital autopilot computer. Additionally, this model requires that missile velocity, V_2, be perfectly estimated and fedback to the guidance computer.

The performance of these four components have been grouped under the construct "autopilot loop". This was done in order to employ a simplified model (Ref 6). This autopilot model assumes that the loop response can be expressed by this formula:

$$\frac{Aout\ (s)}{Acom\ (s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n + {}_n^2} \tag{25}$$

where

Aout(s) = Laplace output acceleration

Acom(s) = Laplace commanded acceleration

ω_n = Undamped natural frequency

ζ = Damping ratio

s = Laplace operator

This assumption is based upon the knowledge that actuator response is much faster than aerodynamic response. The penality of the assumption is that a number of error sources are ignored. Additionally, a number of time delays are lumped under the time lag of the second order response. The final DIS model (Figure 12) incorporates this simplified autopilot loop.

Summary

This chapter presented the model of the DIS missile. This model

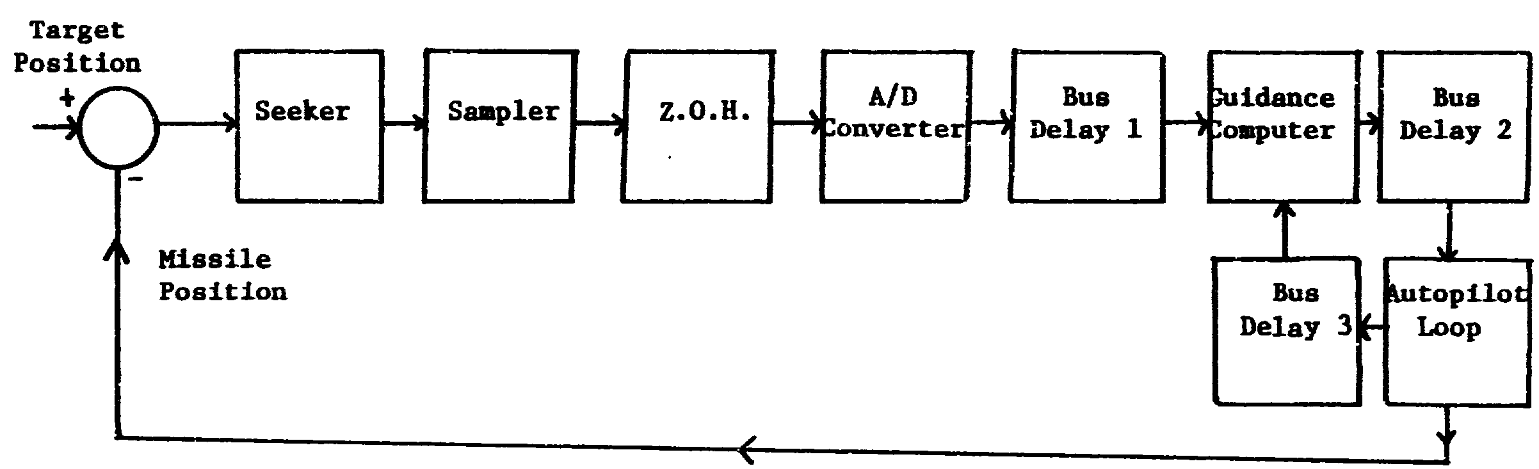


Figure 12. Final DIS Model

included the DIS control loop, the reference frame employed, the major components of the loop, and the simplified autopilot loop. The next step is to explain how the model was incorporated into Tactics IV.

III. Tactics IV Computer Simulation

Introduction

The Tactics IV computer program was used to relate data latency to missile accuracy. This was accomplished by first modifying the program. The Tactics IV missile was made to behave like the DIS missile modeled in Chapter II. Then, a series of experimental cases were run which tested the missile under a variety of conditions. These conditions included a range of delays, engagement profiles, target maneuvers and launch ranges. This test data was compiled and the experimental results are as given in Chapter V.

This chapter begins by introducing the unmodified Tactics IV program. This provides a brief summary of how the simulation is constructed. Next, revised Tactics IV is presented. This includes an explanation of how the final DIS model was transferred to Tactics IV. Finally, the test cases are presented. This section explains what variables were employed in the tests and what output was produced.

Original Tactics IV

The original Tactics IV computer simulation is a Fortran IV program which models an air-air engagement (Ref 6). The program emphasizes flexibility in the setup of the engagement. The user selects the missile and target models to be used, the engagement geometry and a number of other relevant variables. The user inputs are passed to the main program through the input file called Tape 5. The results of the engagement are both sent to the Output file and to a file called Tape 8. The Output file information includes a time history of each engagement and statistical data on "time of closest approach" and "miss distance". The Tape 8 file allows the user to plot a number of engagement variables.

Revised Tactics IV

The revised Tactics IV program is the original program modified so that the missile performs like a DIS missile. This revised program will be explained by first presenting the program flowchart. This will provide the "big picture" of the program. Next, two essential aspects of revised Tactics IV will be introduced. These aspects are the basic integration step size (DT) and the delay model. Then the revised and newly created modules will be summarized. Finally, the Tape 5 input file will be covered.

The revised Tactics IV flowchart is depicted in Fig. 13. The modules called from the main program level are indicated with numbers 1 through 7 in circles. Modules 1 through 4 initialize the Tactics IV engagement. Modules 5 through 7 are located in the "inner loop" of the main program. Policy models missile and target dynamics. This includes the main missile module called "Misilx". All of the revised or new modules reside within Misilx. Module 6 is called Auxcom and its records the progress of the engagement by writing to Output and Tape 8. Finally, Module 7 is called Intgrt. This module advances the missile and target through simulation space over a series of simulation time intervals.

The basic intergration step size (DT) is the time interval over which missile and target motion are integrated. The DIS transmission rate is the frequency at which the DIS computers transmit their data. It was observed that if the integration step size was set equal to the

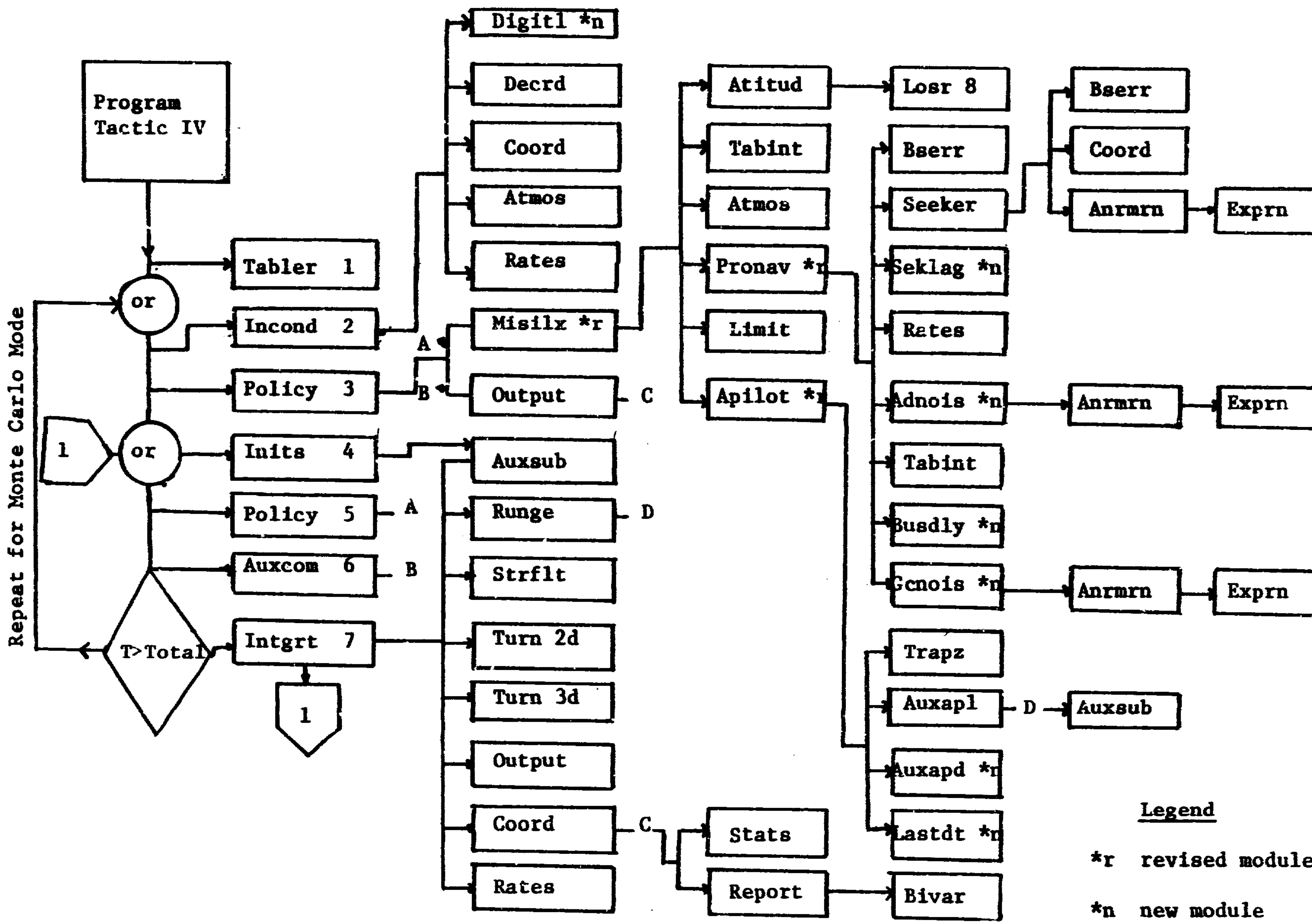


Figure 13. Revised Tactics IV Flowchart

transmission rate then each computer would transmit one message per DT integration interval. The DIS transmission schemes which were examined indicated that transmission frequency varied between 1 and 200 HZ (Ref 11). Two transmission rates were arbitrarily chosen for evaluation. These two rates were 100 HZ and 10 HZ. These two rates were chosen for ease of use. They yield integration step sizes as shown below:

For 100 HZ:

$$DT = \left(\frac{1}{f}\right) = \left(\frac{1}{100}\right)\text{sec} = (.01)\text{sec} \tag{26a}$$

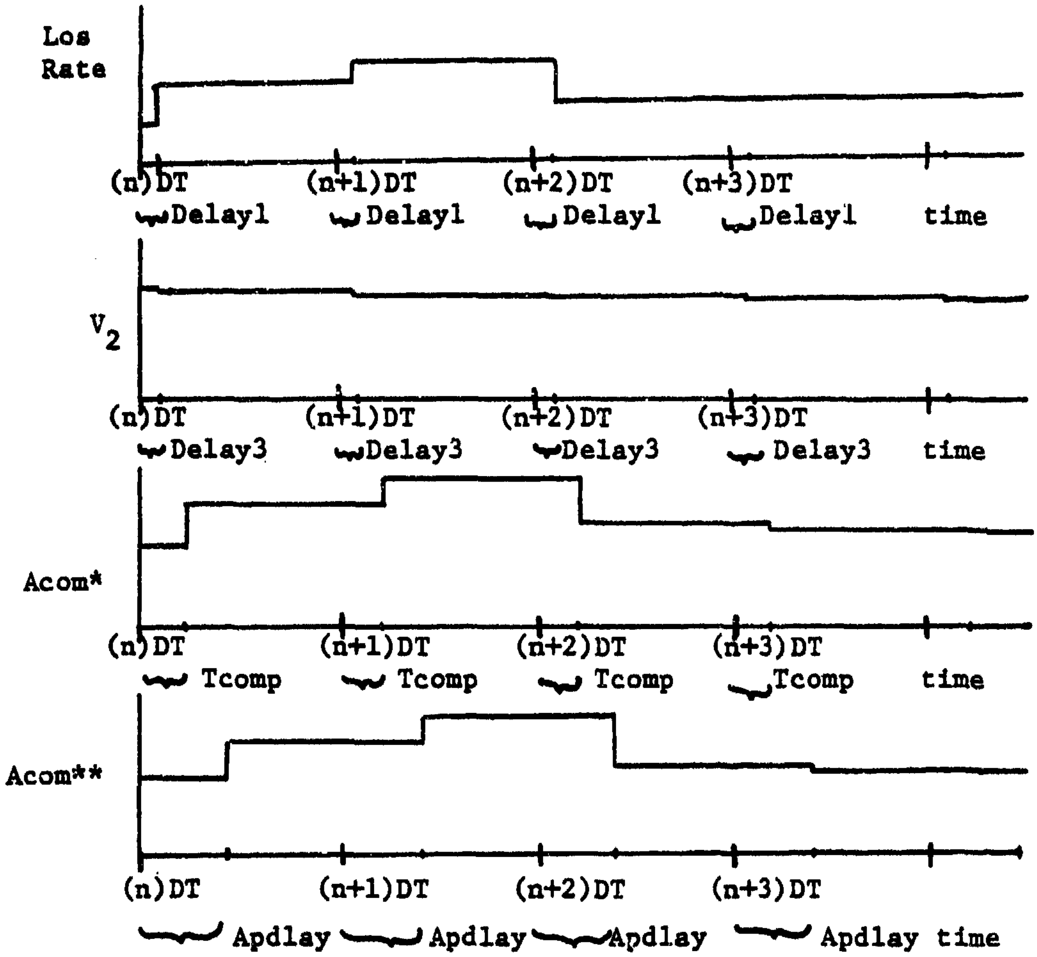
For 10 HZ:

$$DT = \left(\frac{1}{f}\right) = \left(\frac{1}{10}\right)\text{sec} = (.1)\text{sec}$$

The majority of the test data was generated with DT set to (.01)seconds. A small number of computer runs were made with DT set to (.1)seconds and were included to give the study balance. It should be emphasized that the DT integration step size is central to revised Tactics IV. It was particularly essential to the development of the delay model which is introduced next.

The delay model permits the modeling of the DIS delays listed in delays listed in Chapter II. These delays include the seeker's analog-digital converter delay, bus delay 1, computation time delay of the guidance computer, bus delay 2, and bus delay 3. The first aspect of the model is that several of these delays are grouped together. This is done because the grouped delays occur sequentially in the control loop. The first grouping is to call the analog-digital converter delay and bus delay 1 by the name <u>Delay1</u>. The second grouping is to group the guidance computer on delay and bus delay 2 under the title Delay2. Finally, bus delay 3 is termed Delay3. Next, a key assumption is made about the DIS network (See Fig. 12). The assumption is that the LOS rate signal arrives at the analog-digital converter at the exact beginning of every DT integration cycle. Thus, the digital LOS rate arrives at the guidance computer Delay1 seconds into the integration cycle. Similarly, the V_2 velocity information is ready for transmission to the guidance computer at the beginning of every DT integration cycle. Delay3 seconds into the cycle, V_2 arrives at the guidance computer. Next, an assumption is made that the guidance computer performs its computation at a particular time into the DT cycle. This time is referred to as Tcomp. Finally, the commanded acceleration experiences a delay of Delay2 seconds enroute to the autopilot. The sum of Tcomp and Delay2 is the total time into the DT interval which the autopilot loop must wait for the new commanded acceleration. This time is referred to as Apdlay. The three delays (Delay1, Delay2, Delay3) and the computation time (Tcomp) are user input variables.

The interaction of the various delays is best illustrated graphically. In Fig. 14, Delay1 equals .001 seconds. This implies that until .001 seconds into the DT interval, only the last LOS rate data is available at the guidance computer. The new LOS rate data arrives at .001 seconds and is available all the way until .001 seconds into the next DT cycle. Delay3 is also .001 seconds in this example. This implies that until .001 seconds into the DT interval, only the last V_2 missile velocity data is available at the guidance computer. The new V_2 missile velocity data is available .001 seconds into the DT interval. It remains available until .001 seconds into the next DT interval. The third variable is



Notes:

1) <u>Acom*</u> is what Apilot(I) would see if there were no Delay2 lag. Recall, Delay2 is composed of the guidance computer computation delay and the communication delay between the guidance computer and the autopilot.

2) <u>Acom**</u> is the commanded acceleration employed by the Apilot(I) subroutine. Acom** differs from Acom* in that it includes the Delay2 lag. Recall that:

$$\text{Apdlay} = \text{Tcomp} + \text{Delay2}$$

3) Acom* and Acom** <u>are not</u> Tactics IV variables. They only appear here for illustration.

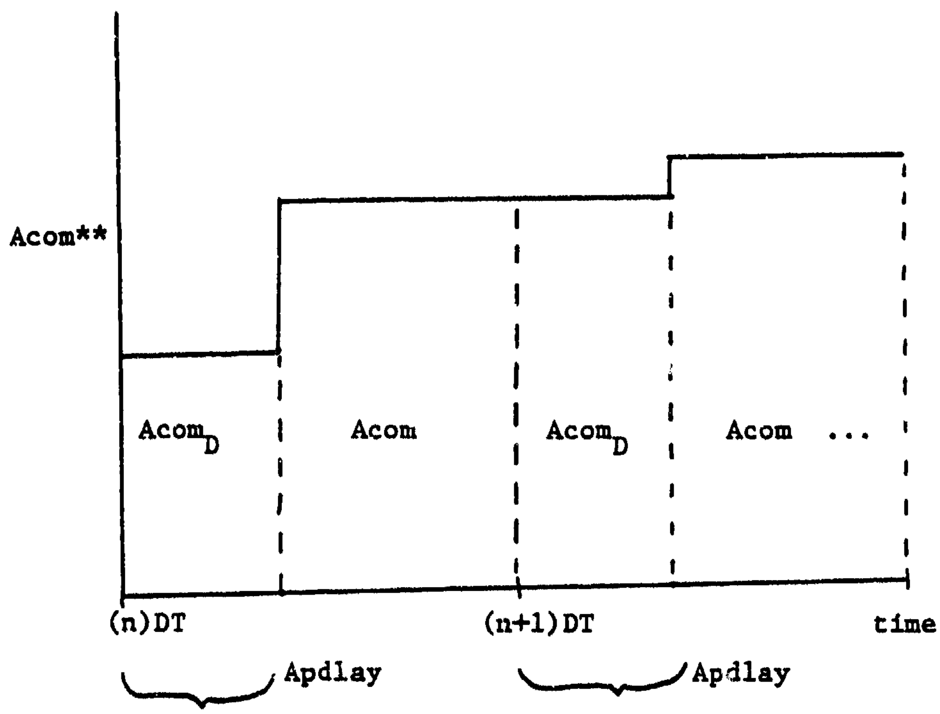
Figure 14. Interaction of DIS Delays

Tcomp and is the time at which the guidance computer performs the proportional navigation computation. Notice that as Tcomp is greater than Delay1 and Delay3, both the new LOS rate data and the new V_2 data are used in the computation. The final delay variable is Delay2. This is the delay between Tcomp and the autopilot loop. Recall that the sum of Tcomp and Delay2 is called Apdlay. Notice that until Apdlay seconds into the DT interval, the autopilot computer uses the last commanded acceleration. The new commanded acceleration does not arrive until Apdlay seconds into the DT interval. This last delay completes the delay model and leads us to a summary of the modules which were altered for revised Tactics IV.

Four of the original Tactics IV modules were altered for revised Tactics IV. These four modules are Apilot (I), Incond, Misilx and Pronav (I). These modules are presented in their entirety in Appendix B. The function of each will be presented here, briefly.

Apilot (I) produces output acceleration based upon a second order response as described in Chapter II. The revised Apilot (I) module does this while allowing for two commanded acceleration levels during the DT time interval. The first commanded acceleration level is called $Acom_D$. This is the commanded acceleration left over from the last DT interval. The second commanded acceleration is called Acom. This is the new commanded acceleration which was delayed enroute to the autopilot. As Fig. 14 and Fig. 15 indicate, $Acom_D$ is used from the start of the DT interval (nDT) until Acom arrives at [(nDT) + Apdlay]. Acom is used from [(nDT) + Apdlay] until the end of the DT interval [(n+1)DT]. At the end of the current DT cycle, Acom becomes $Acom_D$.

The next revised module is Incond. The original routine loads the



Note: As mentioned in Figure 14, Acom** <u>is not</u> a Tactics IV variable. It appears here only for illustration.

Figure 15. Commanded Acceleration used in Apilot(I)

Tape 5 input data array and initializes most Tactics IV variables. The revised module performs the same function including setting several new variables unique to revised Tactics IV. The reader is referred to Appendix B for the specifics.

The third revised module is Misilx. This routine is the master missile module. The only change is added code in support of a new seeker lag routine. The lag is a .10 second delay and the new code permits this lag to run out when the missile goes blind. The seeker lag module is called Seklag and will be explained shortly.

The fourth revised module is Pronav (I). The original routine calls the various seeker modules and computes commanded acceleration based upon the proportional navigation formula. The revised modules do the same while including several new features. This includes a .10 second seeker lag and the injection of analog-digital converter noise. The modules which perform these features, along with the other newly created modules, will be explained next.

Seven new modules were created for revised Tactics IV. These modules are Adnois, Auxapd, Busdly, Digitl, Gcnois, Lastdt and Seklag. The purpose of each will now be presented.

Adnois injects the analog-digital converter noise which the LOS rate experiences prior to bus transmission. This is the zero mean, gaussian noise developed in Section II.

Auxapd assists Apilot (I) in computing output acceleration. It does this by calculating the derivatives of $Acom_D$. These derivatives are used in the trapezoidal integration subroutine which calculates the output accelerations.

Busdly injects the effect of digital delays into the proportional navigation computation. As Fig. 14 indicates, the LOS rate and V_2 data present at the guidance computer at compute time are determined by Delay1, Delay3 and Tcomp. Busdly examines these three variables and insures that the proportional navigation computation is based upon the appropriate data.

Digitl initializes key revised Tactics IV variables. Digitl resides within the Incond module.

Gcnois injects the finite word length noise associated with the guidance computer. This zero mean, gaussian noise is as developed in Section II.

Lastdt assists Apilot (I) in computing output acceleration in the last DT time interval of the engagement. This routine is necessary because the Intgrt routine performs a binary search for the exact time the engagement ended. Intrgt chops the last DT interval into successively smaller "windows" searching for when the missile passed the target. Lastdt calculates how much of the window is Acom and how much is Acom.

Finally, Seklag injects a .10 second time delay into the seeker data transmission. This is accomplished by storing the LOS rate data in an array for .1 seconds of simulation time.

The original Tape 5 input file provided the main program with the necessary parameters for the conduct of the engagement. The Tape 5 file used in revised Tactics IV performs the same function. Additionally, it includes variables unique to revised Tactics IV. These variables are listed in Appendix A.

Fixed Launch Range Tests

The effect of digital delays upon missile accuracy was evaluated through a series of computer runs. The first group of runs involved launch geometries where the target was located a fixed range down the x-axis. These were called the "fixed launch range" tests. Within this test group, several simulation parameters were varied. This included attack geometry, digital delays, target maneuvering, noise modeling (i.e. deterministic or stochastic modeling) and integration step size DT. Each of these variables will now be developed.

The fixed launch range tests involved three geometries. These geometries were the "tail attack", "frontal attack" and "initial heading error attack". The tail attack geometry involved a target heading 45° away from the missile. There was no heading error at launch (See Fig. 16). The frontal attack geometry involved a target heading 45° off the nose of the missile. There was no heading error at launch here either (See Fig 17). In the initial heading error attack geometry, the missile is launched identically to the "tail attack" launch. The target, however, is 1000 ft down the y axis from the tail attack position (See Fig 18).

In the "fixed launch range" tests, the revised Tactics IV delays also were varied. They were varied by applying equal amounts of Tcomp and Delay2. As discussed earlier in this section, the total autopilot delay was called Apdlay. This delay term is computed by the formula:

$$\text{Apdlay} = \text{Tcomp} + \text{Delay2} \qquad (27)$$

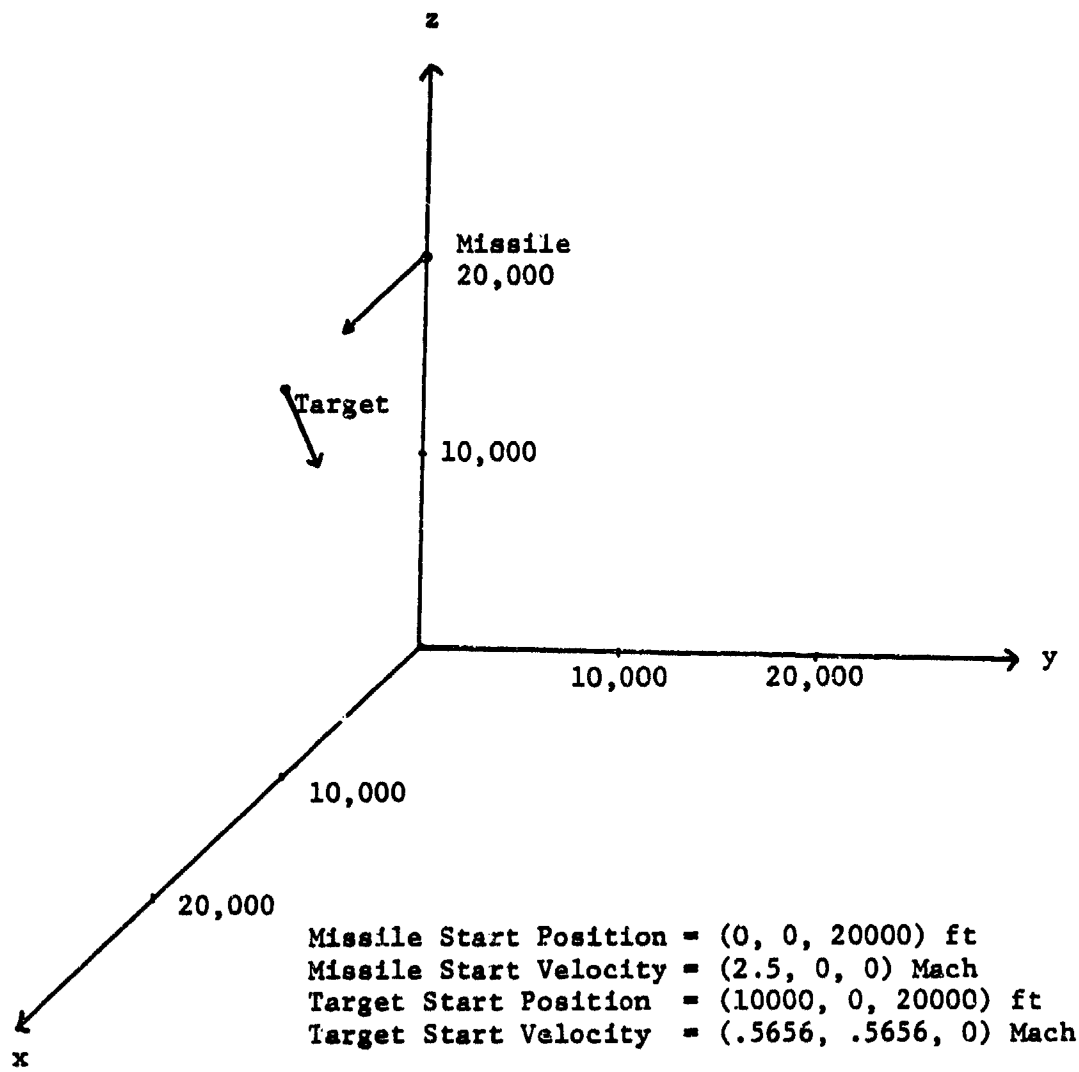


Figure 16. "Tail Attack" Geometry

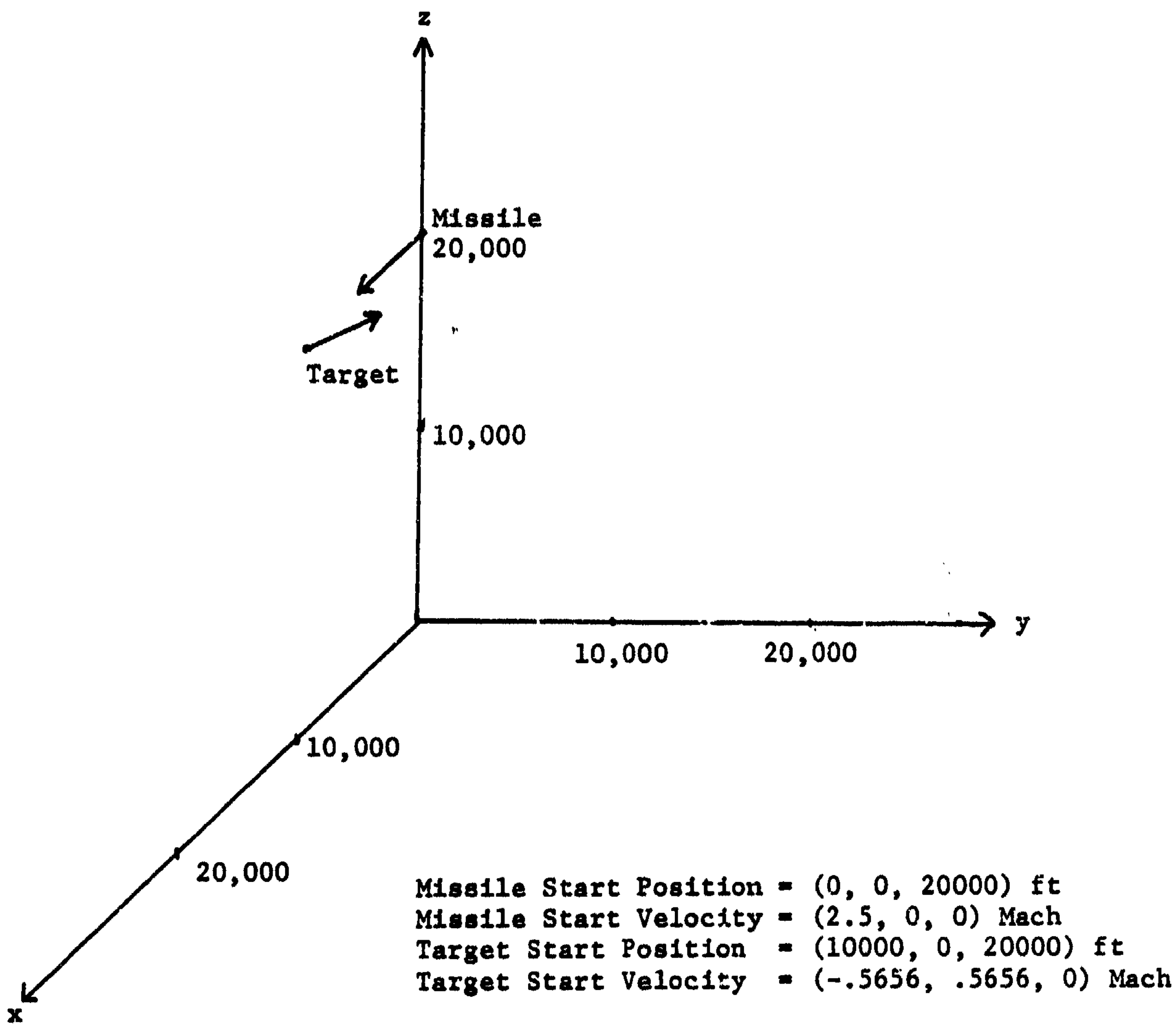
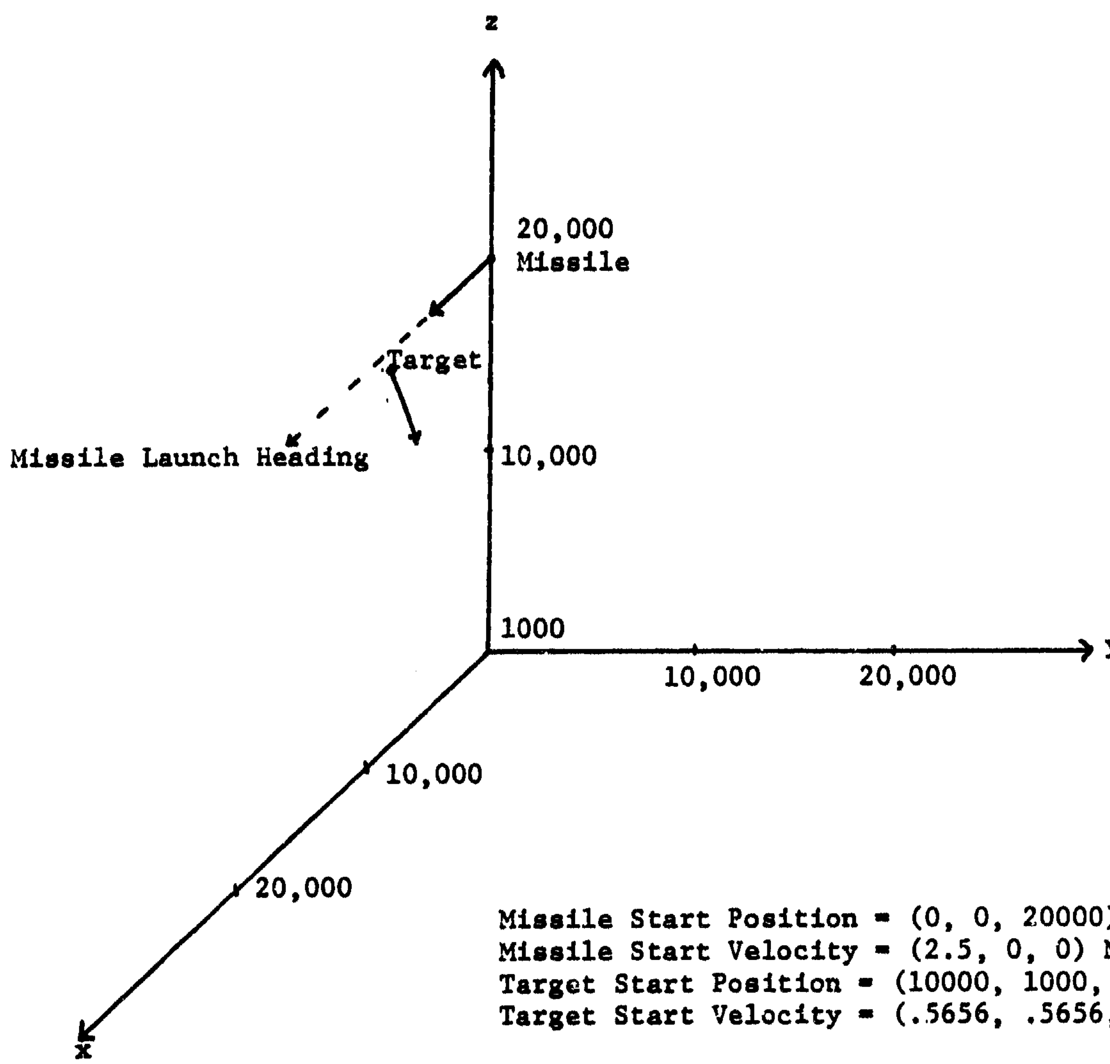


Figure 17. "Frontal Attack" Geometry



Missile Start Position = (0, 0, 20000) ft
Missile Start Velocity = (2.5, 0, 0) Mach
Target Start Position = (10000, 1000, 10000) ft
Target Start Velocity = (.5656, .5656, 0) Mach

Figure 18. "Initial Heading Error Attack" Geometry

where

Tcomp = Computation point of the guidance computer

Delay2 = Delays between guidance computer and autopilot loop

The range on this total delay is as follows:

$$0 \text{ sec} \leq \text{Apdlay} \leq .01 \text{ sec} \tag{28}$$

Five test points were chosen within this range. These are depicted in Table II along with the component Tcomp and Delay2 times.

Also in these tests, three target evasive maneuvers were employed. The first was a straight and level target. The second was a turning target. In this maneuver, the target executes a level, 9 g left turn at 1.0 second time to impact. In the third maneuver, the target climbs and dives at 1.0 second time to impact.

Next, these test runs were conducted both in a deterministic and stochastic mode. The deterministic runs were based upon a missile without stochastic noises. The missile did have the .10 second seeker lag, however. The stochastic missile had all of the stochastic noises and the seeker lag. The stochastic statistics included mean time of flight and mean miss distance. The statistics were based upon 20 trials.

Finally, the basic integration step size DT was set to both .01 and .1 seconds. The .01 seconds test included all of the variables just developed. However, due to time constraints, the .1 seconds test was scaled down to examine only a few of the "fixed launch range" variables. First, the scaled down test examined only the tail attack geometry. This geometry was selected because it is the traditional quadrant for air-air

Table II

Digital Delays for Fixed Launch Range Tests

Delay1 (sec)[1]	Delay2 (sec)[2]	Delay3 (sec)[3]	Tcomp (sec)[4]	Apdlay (sec)[5]
0	0	0	0	0
0	.00125	0	.00125	.00250
0	.00250	0	.00250	.00500
0	.00375	0	.00375	.00750
0	.00500	0	.00500	.01000

1 Delay1 is entered into Tape 5 as Data(81)

2 Delay2 is entered into Tape 5 as Data(82)

3 Delay3 is entered into Tape 5 as Data(83)

4 Tcomp is entered into Tape 5 as Data(84)

5 Apdlay = Delay2 + Tcomp as per equation (28)

missile launches. Next, only three Apdlays were examined instead of five. This was considered an adequate sampling of the Apdlay as the three points spanned the region (i.e. Apldlays of 0, .005, and .01 seconds were chosen). Finally this scaled down test did not employ a reduced number of target maneuvers. Also, both deterministic and stochastic missile models were examined. The difference between the two sets of tests is illustrated in Table III.

Additionally, two types of computer output were produced in the variable launch range test. The first output was a tabular listing of miss distance and time of flight statistics along with "sensor limiting" data (see Chapter IV). The miss distance and time of flight data was graphed versus Apdlay and can be found in Appendix D. The "sensor limiting" data was recorded in the results tables in Chapter V. The second output was computer plots of key engagements variables. These plots included XY versus time, missile acceleration versus time and missile velocity versus time. These plots can be found in Appendix E.

Variable Launch Range Tests

These test runs evaluated the effects of delays upon missile accuracy based upon a variety of target ranges. Constants included the missile launch position and the target aspect angle. Variables included the launch range, target maneuver and missile delays. Deterministic and stochastic runs were both made. Finally, both tabular listings and engagement plots were generated.

The missile position at launch was identical to that in Fig. 16. The target was positioned in a "tail attack" aspect on the x axis and at 20000 ft altitude. The x coordinate was varied between 5000 ft and 15000 ft at

Table III

Fixed Launch Range Test Plan

Launch Range	DT	Attack Geometry	Target Maneuver	Apdlay	Missile Model
10Kft	.01sec	Tail	Str/Lvl	Five test Points*	Deterministic and Stochastic
			Turning	Five test Points*	Deterministic and Stochastic
			Climb/ Dive	Five test Points*	Deterministic and Stochastic
		Frontal	Str/Lvl	Five test Points*	Deterministic and Stochastic
			Turning	Five test Points*	Deterministic and Stochastic
			Climb/ Dive	Five test Points*	Deterministic and Stochastic
		Initial Heading Error	Str/Lvl	Five test Points*	Deterministic and Stochastic
			Turning	Five test Points*	Deterministic and Stochastic
			Climb/ Dive	Five test Points*	Deterministic and Stochastic
	.1sec	Tail	Str/Lvl	Three test Points**	Deterministic and Stochastic
			Turning	Three test Points**	Deterministic and Stochastic
			Climb/ Dive	Three test Points**	Deterministic and Stochastic

* As per Table IV where the five test points are Apdlay = 0, .00250. .00500, .00750 and .010003 seconds.

**As per Table IV where the three test points are Apdlay = 0, .00500 and .01000 seconds.

2500 ft increments. The target was again maneuvered either straight/level, turning, or climb/dive.

The delays were limited to an Apdlay of .00 seconds, .005 seconds and .01 seconds. Only three delays within the region of definition were selected as the larger launch ranges required excessive processing times (See Table IV).

Next, both deterministic and stochastic runs were made. Again, 20 runs were used in the stochastic mode.

Finally, test outputs were again tabular listings of engagement miss distances and computer plots of key variables. The miss distance versus launch range graphs are found in Appendix D. The computer plots are found in Appendix E. A summary of the variable launch range tests is depicted in Table V.

Summary

This completes the presentation of the missile simulation and test plan. This presentation included examining the computer program, developing the DIS delay model and, finally, listing the many variables involved in the test plan. With this presentation complete, only one topic remains before examining the test results. That subject is an examination of the factors which contribute to miss distance.

Table IV

Digital Delays for Variable Launch Range Tests

Delay1 (sec)	Delay2 (sec)	Delay3 (sec)	Tcomp	Apdlay
0	.00000	0	0	0
0	.00250	0	.00250	.00500
0	.00500	0	.00500	.0100

Table V

Variable Launch Range Test Plan

Launch Range	Attack Geometry	Target Maneuver	Apdlay	Missile Model
5KFt	Tail	Str/Lvl	Three test Points*	Deterministic and Stochastic
		Turning	Three test Points*	Deterministic and Stochastic
		Climb/ Dive	Three test Points*	Deterministic and Stochastic
10KFt	Tail	Str/Lvl	Three test Points*	Deterministic and Stochastic
		Turning	Three test Points*	Deterministic and Stochastic
		Climb/ Dive	Three test Points*	Deterministic and Stochastic
15KFt	Tail	Str/Lvl	Three test Points*	Deterministic and Stochastic
		Turning	Three test Points*	Deterministic and Stochastic
		Climb/ Dive	Three test Points*	Deterministic and Stochastic

*As per Table IV, Apdlay = 0 sec, .005 sec, .01 sec

IV. Factors Affecting Miss Distance

Introduction

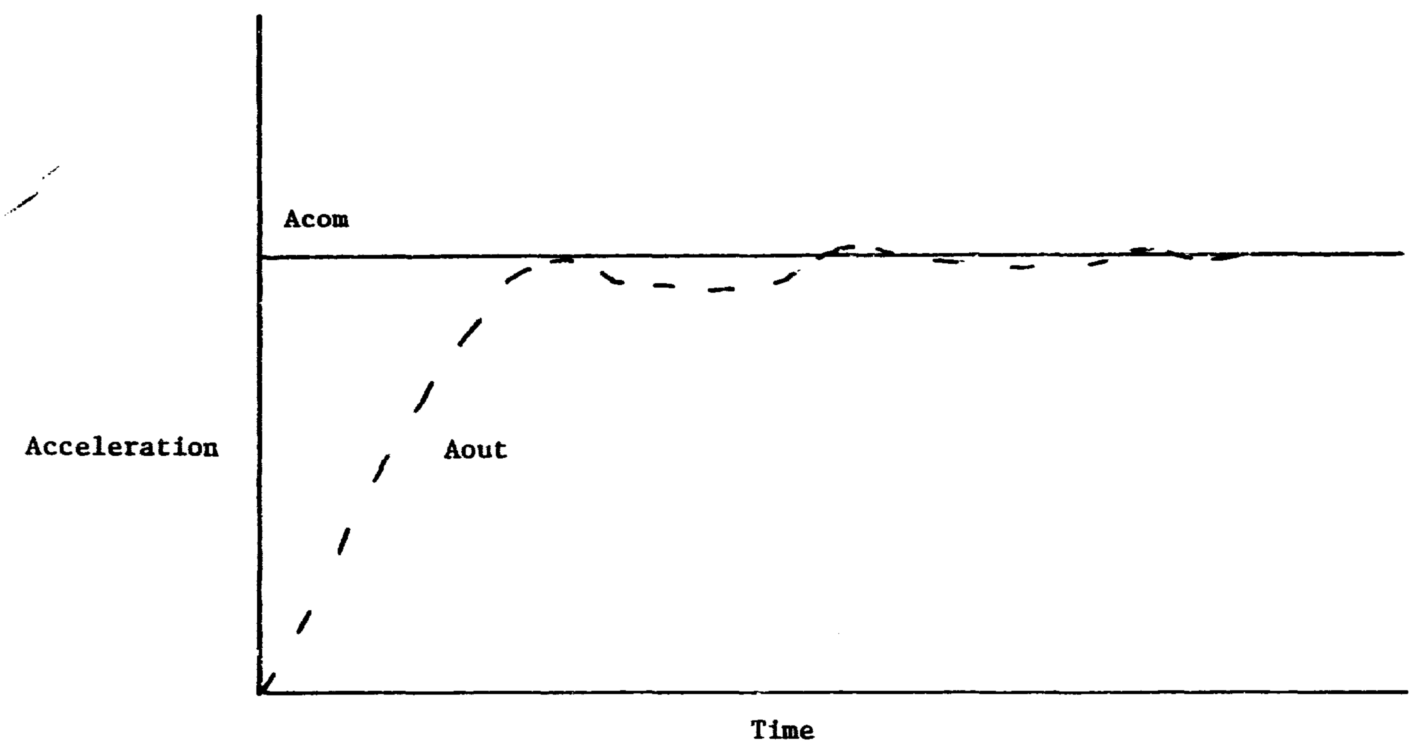
The reasons why a missile fails to impact the target are many. They can include the following: response time, transmission rates, energy loss, g limiting, sensor limiting, sensor noise and target evasive maneuvering. Although this investigation primarily examined the data latency component of response time, a grasp of the other factors is essential. Each of the miss distance factors is addressed separately. This chapter presents each of these factors so as to better explain and support the test results presented in Chapter V.

Response Time

Missile response time is the time required for the missile to achieve commanded acceleration levels. As mentioned, autopilot response is assumed to be second order as per equation (26). Transfer functions of this sort result in time responses as shown in Fig 19 (Ref 1). Notice that the output acceleration rises near the commanded acceleration level and eventually settles at it. The time required for this to occur is called the "settling time", t_s. Factors which define t_s are the autopilot time constant (T_c) and the "lumped" time constant (T_{cl}).

The autopilot time constant is the exponential decay of the autopilot response. The formula which defines the autopilot time constant is as follows:

$$T_c = \frac{1}{\zeta\omega_n} \tag{29}$$



Note: Response based upon $\zeta = .7$

Figure 19. Simple Second Order Response

where

ζ = autopilot damping ratio

ω_n = autopilot undamped natural frequency

This time constant relates to settling time in that settling time is defined as a specified number of time constants. A typical number of time constants is ten (Ref 5). Therefore, 10 time constants after the autopilot receives a commanded acceleration it should have settled at that new level.

The second time response factor, "lumped" time constant, accounts for the effect of the complete DIS guidance loop upon settling time. This constant is merely the sum of all the forward delays of the missile control loop added to the autopilot time constant. This is where the "data latency" delays enter the missile response formula. As these delays can vary over a range of values, the "lumped" time constant varies over a certain range. T_{cl} is computed in Table VI. Employing the $T_{cl\ (min)}$ and $T_{cl\ (max)}$ computed there and the ten time constant standard for settling time produces:

$$t_s = 2.4286 \tag{30}$$

$$t_{s\ (max)} = 2.6286 \tag{31}$$

This explanation of time response is presented to provide the reader with additional insight into missile dynamics. It is incorrect to state that a slower missile response equates to a larger miss distance in <u>all</u> cases. Other studies bear this out as do the test case results in Chapter 5 (Ref 2). For example, Donatelli found that in high altitude, tail chase

Table VI

Lumped Time Constant T_{cl}

Variable	Minimum Value (sec)	Maximum Value (sec)	Remarks
Seklag	.10000	.10000	Fixed lag value
Delayl	.00000	.01000	
Apdlay	.00000	.01000	
T_c	.14286	.14286	Calculated using eq 30 with $\zeta = .7$, $\omega_n = 10$
Totals	.24286*	.26286**	

* This minimum value is referred to as T_{cl}

** This maximum value is referred to as T_{cl}

attacks, a very-responsive missile can miss the target more than a less responsive one. Also, ultra-responsive missiles can be less accurate in a high noise environment. This requires that the designer make compromises between response speed and noise rejection.

Transmission Rates

The rates at which the DIS computers transmit data affect missile performance and thus miss distance. The exact relationship between transmission rates and miss distance is not clear. But, the theorem that theoretically applies is well established. The theorem is "Shannon's sampling theorem." This theorem will first be presented and then applied to the DIS system.

Shannon stated that there is a minimum frequency at which a signal may be sampled if the signal is to be reconstructed (Ref 10). This sampling frequency is analogous to the transmission rate in our DIS loop. this sampling frequency is limited by the following relationship:

$$f_s > 2f_c \tag{32}$$

where

f_s = sampling frequency (cycles/sec)

f_x = highest frequency component of sampled signal (cycles/sec)

Furthermore, Houpis and Lamont indicated that f_s is best set at least eight times greater than f_c. This "pad" allows for noise, data quantization, and system resonant frequencies (Ref 10).

The key question is then, what is the f_c for the DIS control loop? If f_c is assumed to be the undamped natural frequency of the second order autopilot, then f_c is determined as follows:

$$f_c = \frac{\omega_n}{2\pi} \text{ HZ} \tag{33}$$

$$f_c = 1.5915 \text{ HZ} \tag{34}$$

where

ω_n = Autopilot undamped natural frequency of 10 rad/sec

This dictates that f_s be at least eight times f_c or:

$$f_s > 12.732 \text{ HZ} \tag{35}$$

Interestingly, the transmission rate of 100 HZ is safely above this f_s but the 10 HZ rate is slightly below it. What degradation could result from this 10 HZ transmission rate?

Shannon pointed out that violating his theorum results in frequency "folding" or overlap. This folding occurs when the system complimentary frequencies overlap the fundamental frequency spectra. In our case, the autopilot second order frequency response might be distorted at a transmission rate of 10 HZ. Possible results might include sluggish response to commanded levels (rise time, settling time etc). Additionally, the 10 HZ transmission rate system might be more vulnerable to high frequency noise (such as thermal seeker noise).

Thus, the transmission rate is only a miss distance factor if it is below a certain minimum. Shannon's sampling theorem indicates that the 100 HZ rate is well above the minimum. However, the 10 HZ rate is near the minimum and may adversely affect miss distance.

Energy Loss

Missile energy loss relates to miss distance in that insufficient maneuvering energy precludes the missile from hitting the target. This energy loss is caused by drag. The composition of drag will first be presented. Then an example will be used to demonstrate how drag produces energy loss and can affect miss distance.

Drag is composed of parasite drag and induced drag (Ref 11). Parasite drag is the drag caused by the shape of the missile and by turbulent air effects on the missile body. Parasite drag varies directly with the square of velocity (See Fig 20). Induced drag is a side effect of the creation of lift. It varies inversely with square of velocity and directly with angle of attack (See Fig 20). The interaciton of these two drag components are best explained by an example.

In this example, the missile is launched at a range of 15000 ft and at a speed of 2.5 Mach (See Figs. 21 and 22). The target is flying 45° left of the missile's longitudinal axis at launch. Target speed is a constant .8 Mach and the target flight path is straight and level until 1.0 second time to go. At that point, the target executes a left, 9 g evasive turn in an attempt to generate miss distance.

The drag response in this example is best explained by regions. In the region between point A and B, the missile is at maximum speed (assume a "quick boost then long coast" thrust profile). It also is turning left to properly lead the target according to proportional navigation. Parasite drag in this region is at a maximum as missile velocity is maximum. Induced drag during this region is also elevated as the left turn requires an increase in angle of attack. In the region between B and C, the missile is flying "relatively" straight and level (ignoring noise and the line of sight rate due to deceleration). Parasite drag dominates and induced drag is small. Between points C and D, the missile turns hard

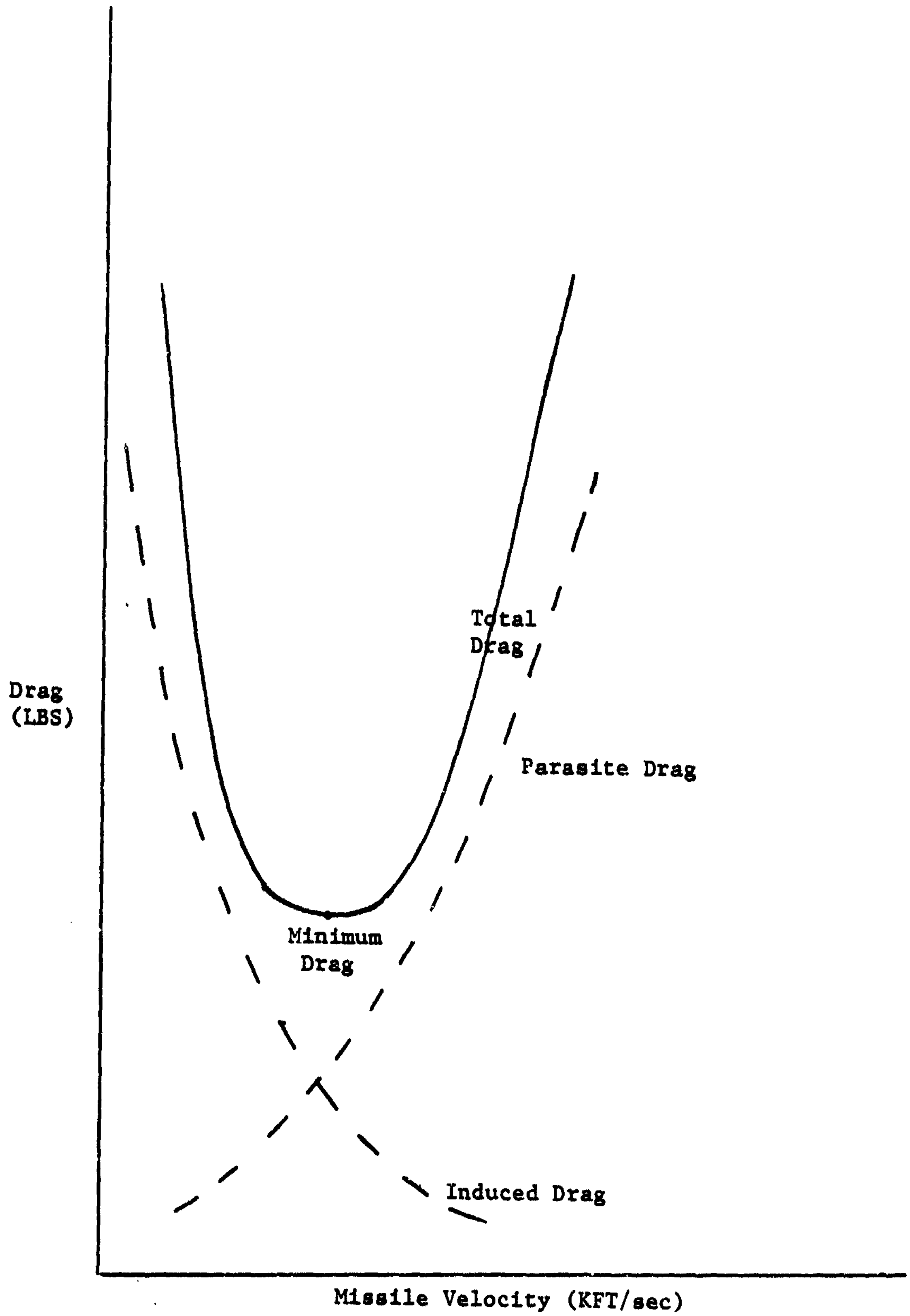


Figure 20. Components of Missile Drag

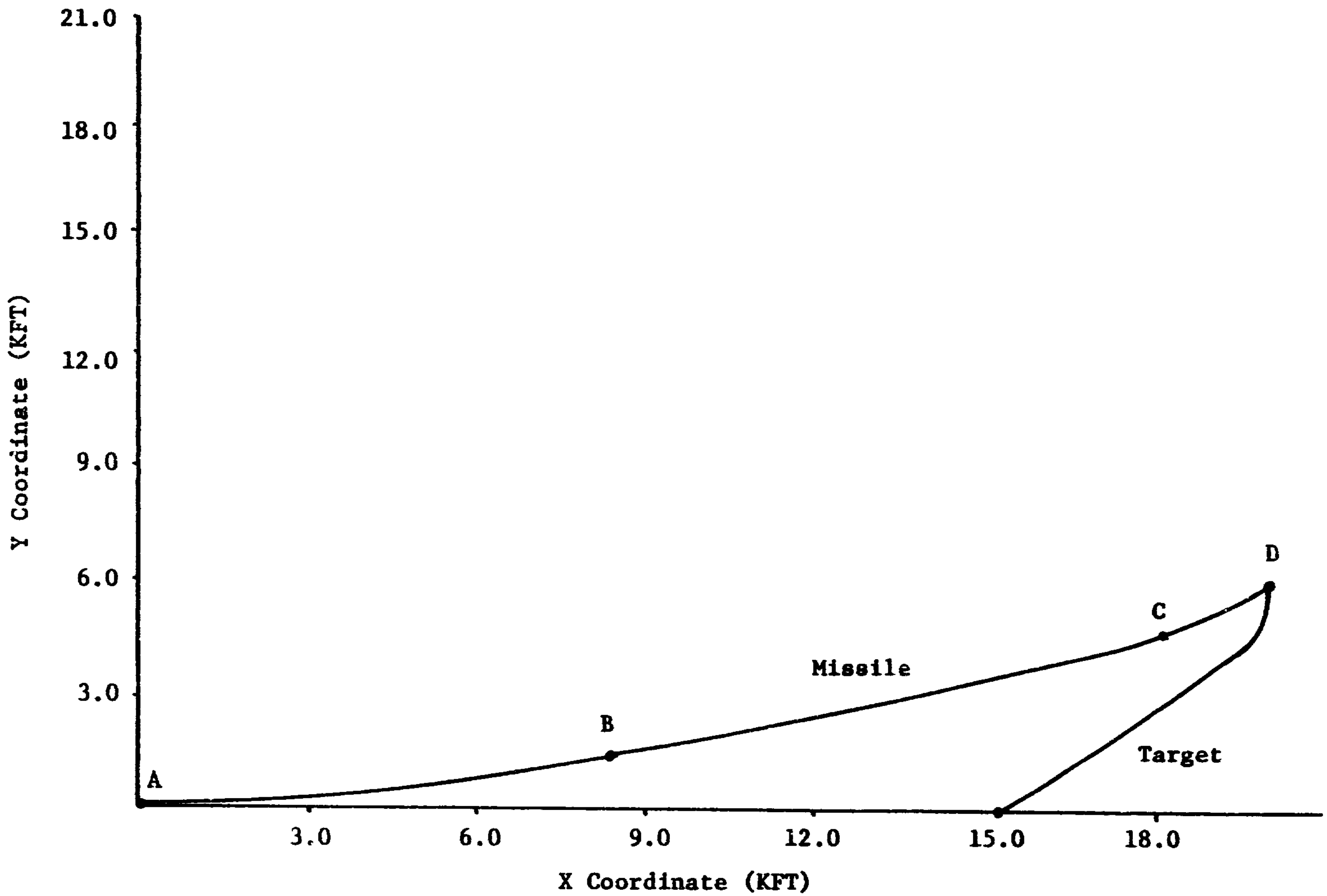


Figure 21. Drag Regions of Missile Engagement

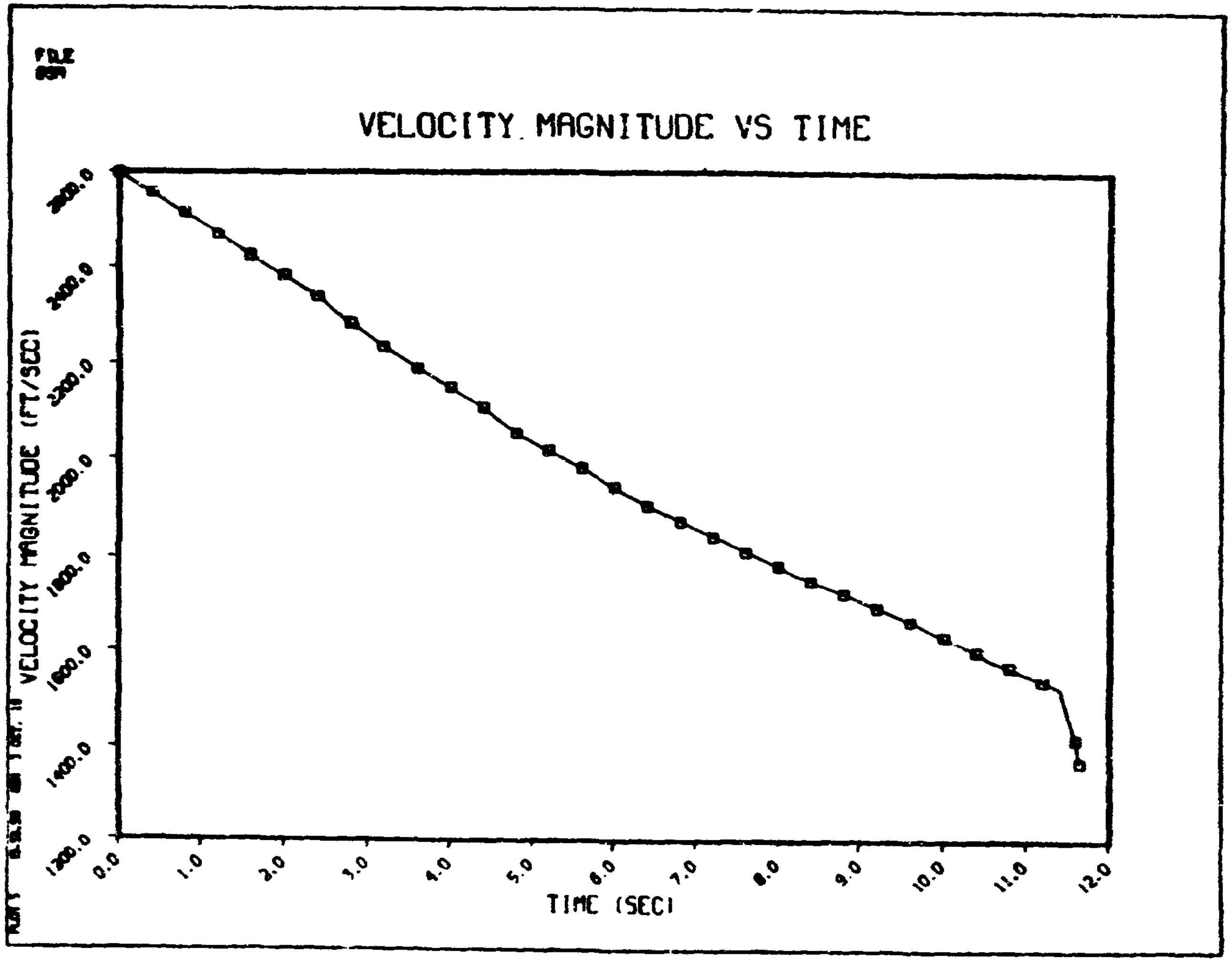


Figure 22. Missile Velocity Profile for 15KFT Tail Attack

left in an effort to null the maneuver induced line of sight rate. Parasite drag still remains a factor. However, the hard left turn requires a large angle of attack. This drives the induced drag up. In this situation, missile velocity begins to drop and additional angle of attack is required for the turn. Eventually, the missile either flys through the "point of closest approach" or stalls.

The effect of drag is to reduce the energy level of the missile. In addition to the obvious fact that a stalled missile can not intercept a target, insufficient energy also affects the sensor. A tail attack missile with a low closing velocity is susceptible to seeker angle limiting (See "sensor limiting" this section). This results in a missile "break lock" condition with the resultant larger miss distance (Ref 2). For the purpose of this study, the missile is assumed to have "insufficient energy" if missile velocity is less than 1.5 times the target velocity.

G Limiting

Missile "g limiting" occurs when the intercept requires more lateral acceleration than the missile can either generate or withstand (Ref 5). The effect of this limiting is that the missile fails to null the line of sight rate and misses the target. In the example previously presented (Figs. 21 and 22), the missile was required to execute a hard, left turn at point C. In order to gain an appreciation of the magnitude of the acceleration the missile must develop, assume that the missile must match the target's turn rate. The target's turn rate is a function of the acceleration achieved and target velocity. The formula is as follows:

$$\dot{\gamma}_3 = \frac{a_3}{V_3} \tag{36}$$

where

a_3 = target achieved lateral acceleration perpendicular to V_3

V_3 = target velocity (magnitude)

Assume a target lateral acceleration of 9 g's, a target velocity of .8 mach and a missile velocity of 2.4 mach. The missile must generate 27 g's to match the target's turn rate. Missile g limits are normally higher than this (40 g's in this study). However, notice that a given target acceleration requires a much higher missile acceleration and in some cases, g limiting does occur.

Sensor Limiting

Missile sensor limiting occurs when the missile is unable to track the target. The effect of this limiting is that the line of sight rate is no longer provided for the guidance computer. Effectively, the missile must fly "blind". This does not imply that the missile completely misses the target. In many missiles, the warhead explodes either upon impact or "point of closest approach".

The first type of sensor limiting is angle limiting. This limiting occurs when the "look" angle of the seeker exceeds a physical limit called field of view ("Look" angle is the resultant angle of the azimuth and elevation angles developed in Chapter 2). The angle limit used in this study is 45 °. Angle limiting occurs most frequently in crossing attack geometries (Ref 2).

The second type of sensor limiting is rate limiting. This limiting occurs when the seeker azimuth or elevation drivers are unable to track the target. The rate limit used in this study is 60°/sec. Rate limiting occurs both in crossing attack geometries and in high noise environments (Ref 2).

The third type of sensor limiting is blind range limiting. This occurs when the missile is so close to the target that the receiver is off when the reflected pulse returns (Ref 2). This range is determined by the following formula:

$$\text{Blind range} = \frac{c}{2\ \text{PRF}} \tag{37}$$

where

c = speed of light

PRF = pulse repition frequency

A typical value for blind range is 100 feet. Blind range limiting occurs in <u>all</u> radar missile intercepts.

The effect of sensor limiting is a function of how long the missile must fly blind before the point of closest approach. This essentially means that the further away from the target angle or rate limiting occurs. the greater the miss distance will be. Blind range limiting should produce greater miss distances for low closing velocity profiles than for high closing velocity profiles.

Sensor Noise

The next miss distance factor is sensor noise. The three sensor noises are radome error, thermal noise and glint. They effect missile accuracy by providing false LOS rates for the guidance computer. The effect of each will now be developed separately.

Radome error is the distortion of the true line of sight which occurs when the radar signal passed through the radome. The equations which describe the error are (10) through (14). The relevant point is that the larger the radar look angle, the greater error. For example, in tail chase and frontal attacks, the look angle and the radome error are small. In beam attacks, look angle and radome error are larger (Ref 2). The result is that as true line of sight rate changes occur, they are distorted by the radome. These incorrect line of sight rates equate to incorrect commanded accelerations and larger miss distances.

Thermal noise is the random noise associated with electrical excitation in the seeker circuitry. This noise is modeled by equations (15) and (16). Notice that the magnitude of the noise is directly proportional to square of the range. Thus, thermal noise is greatest at launch. It's effect is to drain missile energy early in the engagement.

Glint is the apparent wander of the target radar centroid due to phasing errors. It is determined by equations (17) and (18). Notice that glint is inversely proportional to range. Therefore, glint is a minor effect at launch but becomes quite significant at intercept. The effect upon missile accuracy is to produce large, false accelerations late in the engagement. Depending upon missile responsiveness, this can generate large miss distances.

Target Evasive Maneuvering

The final factor affecting missile accuracy is target evasive maneuvering. There is a wide variety of evasive maneuvers but they all attempt to either maximize missile line of sight rate, g limit the missile, exceed sensor limits or stall the missile. The previous examples demonstrated the effect a hard turn can have upon missile accuracy. This study limited target maneuvers to a 9 g level turn and a 3D climb/dive maneuver.

Summary

This chapter introduced the major factors affecting miss distance. These factors included response time, transmission rate, energy loss, g limiting, sensor limiting, sensor noise and target evasive maneuvering. It was also shown that data latency is but a part of the response time factor. With this in mind, the test results are presented next.

V. Results and Recommendations

Introduction

The effect of data latency upon missile accuracy was evaluated through the series of tests outlined in Chapter III. The results of these test were recorded on the graphs of Appendix D and the postprocessor plots of Appendix E. Additionally, the test results were further condensed into Tales VII - XII of this chapter. The information in these seven tables is elaborated on in the following paragraphs:

Results

Table VII

Table VII is the results summary for the fixed launch range, deterministic, DT = .01 sec tests. These tests involved three attack geometries, three target maneuvers and a range of data latency (Apdlay). All of the simulated launches occurred at target x axis ranges of 10000 feet. Additionally, no stochastic noises were simulated in this set of test cases. This was done so as to concentrate on the data latency effects.

The first three rows of Table VII show that data latency (Apdlay) had a mixed effect upon the tail attack cases. The benign target (row 1) actually showed a slight drop in miss distance for increased delays. As the miss distance was so small (less than 1.50 feet), this trend was not deemed significant. The turning target (row 2), showed both larger miss distance and a larger "range" of miss distances for varying Apdlay. Notice that the "maximum" Apdlay (.01 seconds) imparted only an additional

Table VII

Results Summary for Fixed Launch Range, Deterministic, DT = .01 sec Tests

Attack Geometry	Row #	Target Maneuver	Miss Distance Range (ft)	Miss Distance Max Apdlay - No Apdlay (ft)	Other Factors Affecting Miss Distance		
					Energy Loss*	G Limiting**	Sensor Limiting***
Tail	1	Str/Lvl	.28 - 1.50	- .60	No	No	No
	2	Turning	18.50 - 22.67	+ .93	No	No	No
	3	Climb/Dive	29.70 - 35.80	+6.10	No	No	No
Frontal	4	Str/Lvl	2.41 - 2.66	- .205	--	--	No
	5	Turning	14.00 - 16.25	+1.75	--	--	No
	6	Climb/Dive	3.10 - 4.00	+ .90	--	--	No
Initial Heading Error	7	Str/Lvl	.22 - .36	- .14	--	--	No
	8	Turning	34.30 - 39.30	+5.00	--	--	No
	9	Climb/Dive	31.80 - 36.20	+4.4	--	--	No

Legend:
--: Not measured.
*: Considered a factor if V_2 dropped below 1.5 V_3 or 1246.5 ft/sec in this study.
**: G Limiting occurred if commanded acceleration equaled or exceeded 40 g's.
***: Sensor Limiting includes gimbal rate or angle limiting.

.93 feet over the "no" Apdlay condition. The climb/dive target (row 3), showed a larger miss distance then did the first two maneuvers. Furthermore, the largest Apdlay resulted in the largest miss distance. Thus, of the three tail attack maneuvers the climb/dive target was most clearly effected by data latency, the turning target was next and the benign target was unaffected by data latency.

The middle three rows of Table VII show that data latency had a mixed effect upon miss distance. The attack upon benign target (row 4) showed decreasing miss distance for increasing delays. The miss distance remained small though, over the range of Apdlay tested. The turning target (row 5) showed a small increase in miss distance for increased delays. The climb/dive target (row 6), interestingly showed a small miss distance compared to the tail attack (row 3). Maximum data latency produced only .90 feet additional miss distance. Thus, of the three frontal attack maneuvers, Apdlay affected the turning target the most, the climb/dive target next and did not affect the benign target.

The last three rows of Table VII show that data latency again had a mixed effect upon the three target maneuvers. The benign target (row 7) surprisingly showed a smaller miss distance than the simple, tail attack case (row 1). Again, the missile essentially hit the target and Apdlay did not affect that outcome. The turning target (row 8) showed noticeably higher miss distances over the two previous geometries (row 2, 5). Data latency translated directly to an additional 5.00 feet of miss distance. Finally, the climb/dive target (row 9) showed large miss distances, also Data latency directly related to an increased miss distance of 4.40 feet. So of the three maneuvers, data latency strongly affected the turning target and the climb/dive target. It did not affect the straight and level target.

Table VIII

Table VIII is the results summary for the fixed launch range, deterministic, DT = .1 sec tests. These tests involved only the tail attack geometry, three target maneuvers and a range of data latency (Apdlay). Finally, all of the simulated launches occurred at a range of 10000 feet.

The three rows of the table all indicate very large miss distances for a noise free missile. As the table indicates energy loss was a definite factor! The plots in Appendix E indicate that the missile lost a good deal of energy early in the engagement. This prevented proper terminal guidance even against the benign target (row 1). The reason for the large miss distances is apparent when comparing Table VII and Table VIII. The benign target attack resulted in a hit with DT set to .01 seconds (Table VII). The benign target attack resulted in a minimum of 41.27 foot miss with DT set to .1 seconds (Table VIII). Therefore, the transmission rate (which DT represents) was the dominate miss distance factor.

In terms of data latency, the benign target was adversely affected by increasing delays. The turning target was also adversely affected. Finally, the attack against the climb/dive target was improved with increasing delays. However, as the missile missed the target by such a large distance, this trend was not considered significant.

Table IX

Table IX is the results summary for the fixed launch range,

Table VIII

Results Summary for Fixed Launch Range, Deterministic, DT = .1 sec Tests

Attack Geometry	Row #	Target Maneuver	Miss Distance Range (ft)	Miss Distance Max Apdlay - No Apdlay (ft)	Other Factors Affecting Miss Distance		
					Energy Loss*	G Limiting**	Sensor Limiting***
Tail	1	Str/Lvl	41.27 - 79.02	+37.75	Yes	No	No
	2	Turning	46.86 - 67.43	+ 3.93	Yes	No	No
	3	Climb/Dive	85.97 - 94.92	- 8.09	Yes	No	No

Legend: *: Considered a factor if V_2 dropped below 1.5 V_3 or 1246.5 ft/sec in this study.
**: G Limiting occurred if commanded acceleration equaled or exceeded 40 g's.
***: Sensor Limiting includes gimbal rate or angle limiting.

Table IX

Results Summary for Fixed Launch Range, Stochastic, DT = .01 sec Tests

Attack Geometry	Row #	Target Maneuver	Miss Distance Range (ft)	Miss Distance Max Apdlay - No Apdlay (ft)	Other Factors Affecting Miss Distance		
					Energy Loss*	G Limiting**	Sensor Limiting***
Tail	1	Str/Lvl	36.2 - 39.8	+3.60	No	Yes	Gimbal Rate
	2	Turning	72.45 - 73.63	+1.15	No	Yes	Gimbal Rate
	3	Climb/Dive	92.15 - 94.95	+2.80	No	Yes	Gimbal Rate
Frontal	4	Str/Lvl	27.75 - 27.95	+ .20	--	---	Gimbal Rate
	5	Turning	61.25 - 63.45	+2.20	--	---	Gimbal Rate
	6	Climb/Dive	74.0 - 74.10	+ .10	--	---	Gimbal Rate
Initial Heading Error	7	Str/Lvl	43.2 - 46.4	-3.20	--	---	Gimbal Rate
	8	Turning	89.3 - 92.1	+2.80	--	---	Gimbal Rate
	9	Climb/Dive	85.4 - 88.0	+1.10	--	---	Gimbal Rate

Legend: ---: Not measured.
*: Considered a factor if V_2 fropped below 1.5 V_3 or 1246.5 ft/sec in this study.
**: G Limiting occurred if commanded acceleration equaled or exceeded 40 g's.
***: Sensor Limiting includes gimbal rate or angle limiting.

deterministic, DT =.1 sec tests. These tests are identical to those presented in Table VII except stochastic noises were included. The results depicted in Table IX are elaborated in the following paragraphs.

The first three rows of Table IX show how much greater the miss distances are for the stochastic missile model (See Table VII, rows 1 through 3). It is also interesting that all three tail attack maneuvers indicate an increase in miss distance for increasing data latency. The benign target (row 1) showed a moderate increase in miss distance for increasing delay (3.60 feet). The turning target (row 2) showed much larger miss distances than the benign target. However, it showed smaller increases in miss for increasing Apdlay (1.15 feet). The climb/dive target (row 3), showed very large miss distances and a moderate increase in miss for maximum delay (2.80 feet). Thus, of the three tail attack maneuvers, the benign target was most affected by the delays, the climb/dive target was next and the turning target was least affected. Additionally, notice that two other factors significantly affected miss distance in rows 1, 2 and 3. These two factors were g limiting and gimbal rate limiting.

The middle three rows of Table IX indicate that data latency only slightly affected the three frontal attack maneuvers. The benign target (row 4) miss distance only increased .20 feet for maximum delay. The turning target (row 5) increased the most at 2.20 feet. The climb/dive target (row 6) was essentially unaffected by Apdlay. Additionally, it should be noted that gimbal rate limiting affected all three of these results.

The final three rows of Table IX indicate the data latency had a mixed affect upon miss distance for the initial heading error launches. The benign target (row 7) showed large miss distances for a non maneuvering target. Additionally, it showed a decrease of 3.20 feet miss distance for maximum data latency. The turning target showed an increase of 2.80 feet for the maximum Apdlay. The climb/dive target showed an increase of 1.10 feet for maximum Apdlay. Thus, the turning target was most affected by Apdlay, the climb/dive target was next and data latency improved accuracy for the benign target.

Table X

Table X is the results summary for the fixed launch range, stochastic, DT = .1 sec tests. These tests involved only the tail attack geometry, three target maneuvers and a range of data latency (Apdlay). Again, only launch ranges of 10000 feet were considered.

The three rows of Table X all indicate very large miss distances. The table indicates that both energy loss and sensor limiting contributed to the miss distance. However, comparing Table X with the first three rows of Table IX, the affect of transmission rate is evident again. The identical engagements result in significantly higher miss distances for DT = .1 over DT = .01 seconds. Thus, transmission rate is the dominant miss distance factor.

The effect of data latency upon accuracy was to degrade accuracy. Maximum data latency injected additional miss distances of 7.44 feet for the benign target, 6.80 feet for the turning target and 6.92 feet for the climb/dive target. This affect was not considered as important as the fact that miss distance was large due to transmission rate.

Table X

Results Summary for Fixed Launch Range, Stochastic, DT = .1 sec Tests

Attack Geometry	Row #	Target Maneuver	Miss Distance Range (ft)	Miss Distance Max Apdlay - No Apdlay (ft)	Other Factors Affecting Miss Distance		
					Energy Loss*	G Limiting**	Sensor Limiting***
Tail	1	Str/Lvl	148.46 - 155.90	+7.44	Yes	No	Gimbal Rate
	2	Turning	171.60 - 178.41	+6.80	Yes	No	Gimbal Rate
	3	Climb/Dive	148.14 - 155.06	+6.92	Yes	No	Gimbal Rate

Legend:
*: Considered a factor if V_2 dropped below 1.5 V_3 or 1246.5 ft/sec in this study.
**: G Limiting occurred if commanded acceleration equaled or exceeded 40 g's.
***: Sensor Limiting includes gimbal rate or angle limiting.

Table XI

The test senario for Table XI was similar to Table VII except that the launch range varies between 5000 feet and 15000 feet. Additionally, only the tail attack, deterministic missile was considered.

The 5K launches (rows 1 through 3), demonstrate a mixed relationship between Apdlay and miss distance. The benign target (row 1) showed larger miss distance than the 10K or 15K launches (rows 4 and 7). The maximum data latency produced an additional 1.90 feet of miss distance. The turning target was essentially unaffected by data latency (decreased .70 feet for maximum Apdlay). The climb/dive target showed a relatively large increase in miss distance for maximum Apdlay (6.00 feet). Thus, the climb/dive target was most affected by data latency. The benign target was slightly affected and the turning target was unaffected.

The 10K launches (rows 4 through 7) are identical to rows 1 through 3 of Table VII. Notice here, however, that the 10K launch range resulted in smaller miss distance for the benign and the climb/dive target over the 5K launches. The turning target showed an increase in miss distance over the 5K launches.

The 15K launches (rows 7 through 9) show a mixed relationship between data latency and miss distance. The benign target was hit and delays did not affect accuracy. The turning target (row 8) showed a slight decrease in miss distance for increasing Apdlays (1.00 feet). The climb/dive target (row 9) showed a large increase in miss distance for increasing Apdlay (10.00 feet). Thus, the climb/dive maneuver was most affected by data latency at 15K, the benign target was unaffected and the turning target accuracy was slightly improved by data latency.

Table XI

Results Summary for Tail Attack, Variable Launch Range, Deterministic, DT = .01 sec Tests

Range	Row #	Target Maneuver	Miss Distance Range (ft)	Miss Distance Max Apdlay - No Apdlay (ft)	Other Factors Affecting Miss Distance		
					Energy Loss*	G Limiting**	Sensor Limiting***
5K	1	Str/Lvl	4.90 - 6.80	+ 1.90	No	No	No
	2	Turning	13.20 - 13.80	- .70	No	No	No
	3	Climb/Dive	34.0 - 40.2	+ 6.00	No	No	No
10K	4	Str/Lvl	.28 - 1.50	+ .80	No	No	No
	5	Turning	18.50 - 23.67	+ 1.00	No	No	No
	6	Climb/Dive	29.7 - 35.80	+ 6.50	No	No	No
15K	7	Str/Lvl	.40 - .50	- .10	No	No	No
	8	Turning	22.80 - 23.80	- 1.00	No	No	No
	9	Climb/Dive	25.5 - 35.5	+10.0	No	No	No

Legend: *: Considered a factor if V_2 dropped below 1.5 V_3 or 1246.5 ft/sec in this study.
**: G Limiting occurred if commanded acceleration equaled or exceeded 40 g's.
***: Sensor Limiting includes gimbal rate or angle limiting.

Table XII

The final tests were the variable launch range, stochastic runs summarized in Table XII. These tests were similar to those in Table XI. except that the stochastic missile model was used.

The 5K results (rows 1 through 3) again indicate significantly greater miss distance for the stochastic missile over the deterministic (see Table IX rows 1 through 3). The benign target was unaffected by data latency. The turning target showed an increase of 3.00 feet for maximum data latency. The climb/dive target showed a similar increase fo 2.50 feet for maximum Apdlay.

The 10K results (rows 4 through 7) are as presented in rows 1 through 3 of Table VIII. Notice that the miss distances are the same for the 5K and 10K targets (rows 1 through 4). They are essentially the same for the turning targets (rows 2 and 5). Finally, the miss distance is slightly higher for the 10K climb/dive target over the 5K climb/dive target (rows 6 and 3).

The 15K launches indicate an interesting result. Maximum data latency <u>decreases</u> miss distance by 7.00 feet for the benign target! It should be noted that two factors other than response time significantly affected accuracy (row 7). These two factors were energy loss and gimbal rate limiting. The turning target (row 8) had a very large miss distance (94.5 feet). Accuracy was not affected by data latency. The climb/dive target also had a very large miss distance (100 feet). Accuracy was also unaffected by data latency.

Summary

The overall effect of data latency upon accuracy is summarized in

Table XII

Results Summary for Tail Attack, Variable Launch Range, Stochastic, DT = .01 sec Tests

Range	Row #	Target Maneuver	Miss Distance Range (ft)	Miss Distance Max Apdlay - No Apdlay (ft)	Other Factors Affecting Miss Distance		
					Energy Loss*	G Limiting**	Sensor Limiting***
5K	1	Str/Lvl	37.0	0.00	No	No	Gimbal Rate
	2	Turning	78.0 - 81.0	+3.00	No	No	Gimbal Rate
	3	Climb/Dive	79.0 - 80.3	+2.50	No	No	Gimbal Rate
10K	4	Str/Lvl	36.2 - 39.8	0.00	No	Yes	Gimbal Rate
	5	Turning	72.45 - 73.63	0.00	No	Yes	Gimbal Rate
	6	Climb/Div	92.15 - 94.95	±3.00	No	Yes	Gimbal Rate
15K	7	Str/Lvl	53.0 - 61.5	-7.00	Yes	No	Gimbal Rate
	8	Turning	94.5	0.00	No	No	Gimbal Rate
	9	Climb/Div	100.2 - 101.0	0.00	Yes	No	Gimbal Rate

Legend: *: Considered a factor if V_2 dropped below 1.5 V_3 or 1246.5 ft/sec in this study.
**: G Limiting occurred if commanded acceleration equaled or exceeded 40 g's.
***: Sensor Limiting includes gimbal rate or angle limiting.

Table XIII. This Table was compiled based upon the fixed range statistics of Tables VII through X. The quantity averaged is the amount of additional miss the maximum Apdlay injects over the zero Apdlay miss distance. Notice that the deterministic, DT = .01 second cases indicate an average increase of only 2.12 feet. The stochastic, DT = .01 second cases show an even smaller increase of 1.19 feet. This table also indicates that data latency increased miss distance for DT = .1 seconds. The increases were larger than for the DT = .01 second cases. Data latency degraded accuracy an average of 11.2 feet for the deterministic missile and 7.05 feet for the stochastic missile.

In conclusion, the effect of data latency upon missile accuracy is small relative to other factors (e.g. noises). Many variables determine whether the delays will increase the miss and by how much. For DT = .01 seconds, the greatest increase in miss distance for a data latency of .01 seconds was 10 feet. This occurred with a large launch range, deterministic missile facing a climb/dive target (Table IX, row 9). The typical increase was smaller, however. The 10000 foot launch range scenarios showed an average deterministic miss of 2.12 feet and a stochastic miss of 1.14 feet. For DT = .1 seconds, the missile missed the target due to the low transmission rate. With this in mind, data latency definitely affected miss distance. The maximum increase was for a benign target, deterministic missile (+37.75 feet). Average increase in miss distance was 11.2 feet deterministic and a stochastic miss of 7.05 feet.

Recommendations

This study used a particular missile model, inserted it into a

Table XIII

Average Increase in Miss Distance

DT	Target Maneuver	Average Increase in Miss Distance (ft)*	
		Deterministic*	Stochastic**
.01 sec	Str/Lvl	- .315	+ .20
	Turning	+ 2.56	+2.05
	Climb/Dive	+ 3.80	+1.33
	All Maneuver Avg.	+ 2.12	+1.19
.1 sec	Str/Lvl	+37.75	+7.44
	Turning	+ 3.93	+6.80
	Climb/Dive	- 8.09	+6.92
	All Maneuver Avg.	+11.2	+7.05

*Average amount of additional miss which maximum data latency produced over the no delay miss distance. Plus signs indicate a degradation in accuracy while minus signs indicate an improvement in accuracy.

computer simulation and generated miss distance statistics. The two areas for improvement include the missile model and the computer simulation.

The missile model employed a second order autopilot. Future work should model the three separate components of the autopilot. These components are the digital autopilot computer, the actuator computer and the inertial reference computer. This change would produce a more accurate DIS missile model.

The Tactics IV simulation requires two changes. The first change is to include a variable transmission rate. This implies that each microcomputer in the loop should transmit at its own unique rate. These rates should be user selectable and in the range of .2 to 200 HZ. The other Tactics IV change is to employ a 6 DOF simulation. This option is currently being inserted into Tactics IV by AFWAL/FIA, Wright Patterson AFB, Ohio.

Bibliography

1. D'Azzo, J. J. and Houpis, C. H. Linear Control System Analysis and Design. New York: Mc Graw - Hill Co., 1981.

2. Donatelli, G. A. "Effect of 1995 Missile and Target parameters on Mis Distance". Unpublished thesis. Wright Patterson Air Force Base, OH: Air Force Institute of Technology, 1983.

3. Franklin, G. F. and Powell, J. D. Digital Control of Dynamic Systems. Reading, MA: Addison - Wesley Publishing Co., 1980.

4. Garnell, P. and East, D. J. Guided Weapon Control Systems. Oxford, England: Pergamon Press, 1977.

5. Grote, R. S. and Hutcheson, J. H. Aircraft Survivability Conceptual Design Criteria, AFWAL-TR-82-3062, Vol I. Wright Patterson AFB, OH: Air Force Wright Aeronautical Laboratories, 1982.

6. Grote, R. S. and Hutcheson, J. H. Aircraft Survivability Conceptual Design Criteria, AFWAL-TR-82-3062, Vol II. Wright Patterson AFB, OH: Air Force Wright Aeronautical Laboratories, 1982.

7. Henne, A. M. and Geyer, G. W. "Intergration of Digital Aironics Components for Guided Weapons", NATO AGARD-CP-292, Eglin AFB, FL, May, 1980.

8. Henne, A. M. "Error Analysis of Distributed Control Systems", AFATL/DLMM, Eglin AFB, FL, Mar, 1983.

9. Hlavaty, C. W. "A Practical Three Dimension, II State Extended Kalman Filter for Use in a Fire Control System Against Non-Thrusting Missiles", Vol I. Unpublished Thesis. Wright patterson Air Force Base, OH: Air Force Institutue of Technology, Dec, 1976.

10. Houpis, C. H. and Lamont, G. Digital Control Systems (Theory, Hardware, Software), New York: McGraw - Hill Co., 1983 (Copyright pending).

11. Kershner, W. K. Advanced Pilot's Flight Manual, Ames, Iowa: Iowa State University Press, 1970.

12. Ventre, A. J., Meadows, J. F. and Anderson, W. F. "Effects of Sample Rate and Data Latency on Digital Autopilots", Vol I, Analysis Summary", AFATL-TR-81-37, Eglin AFB FL, April, 1981.

Appendix A

Tape 5 Input File

Tactics IV is a flexible, air-air missile simulation. This flexibility is provided by the Tape 5 input file. This file permits the user to initialize the engagement, specify the missile model, dictate target maneuvering and select a number of other options. The file consists of two main sections. The first section describes the missile's aerodynamic characteristics. The second section defines the other engagement parameters. These two sections are described below along with a sample Tape 5 listing.

The aerodynamic characteristics of the missile are specified by the value of certain key aerodynamic coefficients. These coefficients are tabulated for a range of Mach numbers. The format is as shown in Table A-I. The values are those that describe an unclassified generic missile. These figures were provided by the Air Force Armament Laboratory, AMRAAM section. Additionally, if the user wishes to employ a time varying navigation constant, it is entered here. Table A-I shows a non-time varying navigation constant.

The other engagement parameters are read from Tape 5 into the Data array of Tactics IV. The 100 input data elements are shown in Table A-II. Notice the values assigned for this study. Also, notice that Data elements 80 through 85 were used exclusively in this study.

Finally, a sample Tape 5 listing was included. Notice that the top section is the missile aerodynamics data. Next, there is a comment line. Finally, the data elements are listed. The data is organized five elements to the row. In those cases where no data is entered for a given element, Tactics IV will either set the element to zero or to a standard default value. The reader is referred to reference (Tactics IV, Vol II) for further details on Tape 5.

Table A-I

Missile Aerodynamic Data

Mach	0.20	0.80	1.50	2.00	2.35	2.87	3.95	4.60
$(C_{DO})_{BOOST}$	0.235	0.240	1.05	0.910	0.830	0.745	0.630	0.580
$(C_{N\alpha})$	1.04	1.04	1.04	0.93	0.86	0.90	0.90	0.87
$(C_{M\sigma})$	0.755	0.755	0.775	0.413	0.288	0.180	0.108	0.090
α/σ	0.93	0.93	0.93	0.64	0.62	0.42	0.33	0.31
$(\Delta C_{DO})_{COAST}$	0.116	0.127	0.198	0.162	0.141	0.113	0.070	0.051
Time	0.0	4.0	6.0	8.0	10.0	12.0	14.0	16.0
λ	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0

Table A-II

Input Data Elements

Data No.	Program Variable	Assigned Value	Description
1	R(2,1)	0	Missile x-coordinate
2	R(2,2)	0	Missile y-coordinate
3	R(2,3)	20000	Missile z-coordinate
4	V(2.4)	2.5	Missile velocity vector magnitude. See Data (13)
5	V(2,5)	0	Angle measured in horizontal plane, missile
6	V(2,6)	0	Flight path angle measured in vertical plane, missile
7	R(3,1)	(1)	Target x-coordinate
8	R(3,2)	(2)	Target y-coordinate
9	R(3,3)	20000	Target z-coordinate
10	V(3,4)	.8	Target velocity vector magnitude vehicle 3. See Data (13)
11	V(3,5)	(3)	Angle measured in horizontal plane, target
12	V(3,6)	0	Flight path angle measured in vertical plane, target

*Units are distance (ft), time (sec), velocity (ft/sec or Mach no.), acceleration (g's), angles (deg), weight (lb), area (ft^2).
Default values are listed in text.

(1) 10000 for the fixed launch range test. Varies between 5000 and 15000 for the variable launch range test.
(2) 0 for the tail and frontal attack, 1000 for the initial heading error attack.
(3) 45 for the tail and initial heading error attacks, 135 for the frontal attack.

Table A-II (Cont'd)

Data No.	Program Variable	Assigned Value	Description
13	JATMOS	1	Flag for reading in velocity magnitudes (DATA 4, DATA 10) JATMOS = 0 for ft/sec JATMOS = 1 for Mach number
14	TBURNI	1 (1)	First stage rocket motor burning time
15	TBURN2	1 (1)	Second stage rocket motor burning time
16	WO(2)	370	Missile initial weight
17	BRNR81	1	First stage rocket motor burn rate (lbs/sec)
18	BRNR82	1	Second stage rocket motor burn rate (lbs/sec)
19	CLMAX(2)	0	Maximum aerodynamic lift coefficient missile
20	ASMAX(2)	40	Maximum lateral acceleration limit (structural) missile
21	TGUIDE	0	Time interval that missile is to fly unguided after launch
22	LAMBDA	4	Navigation constant for guidance (missile)
23	TAU(1)	0	First order exponential time lag vehicle 1, (if used)
24	TAU(2)	0	First order exponential time lag missile
25	TAU(3)	0	First order exponential time lag target
26	TAU1	.01	Lead term in autopilot transfer function, APILOT subroutine
27	TAU2	.01	Lag term in autopilot transfer function, APILOT subroutine

(1) Missile thrust modeled as instantly at maximum.

Table A-II (Cont'd)

Data No.	Program Variable	Assigned Value	Description
28	OMEGAN	10	Natural frequency in autopilot transfer function, APILOT subroutine (rads/sec)
29	ZETA	.7	Damping factor of autopilot transfer function, APILOT subroutine
30	NINT	100	Inner "DO Loop" cycle rate
31	SLOPE	1.0E+20	Slope of C_L or C_N versus α curve if assumed to be a constant
32	CDO	0	Zero or profile drag coefficient in either wind or body axis system
33	AREA(2)	.349066	Aerodynamic reference area, missile (ft^2)
34	MANUVR	(1)	Flag defining target maneuver = 0 Straight flight = 1 Turn, left or right as defined by DATA (46) = 2 Calls SMART target maneuver (turning, diving, climbing using DATA (35) and DATA (50) = 3 Calls dive maneuver (man be used for Split S dive/climb) = 4 Calls for 3D climbing/diving turn as defined by DATA (46) and DATA (47) = 5 Calls for MAXACC maneuver to maximize missile turning acceleration = 6 Calls for barrel roll maneuver using DATA (63) and DATA (64) = 7 Calls for jinking maneuver using DATA (65)

(1) 0, 1, and 4 used.

Table A-II (Cont'd)

Data No.	Program Variable	Assigned Value	Description
35	TGO	1	Time-to-go before point of closest approach (set DATA (50) to large value if used)
36	IPOST	1	Flag option for writing binary data file for plot post processor = 0 No file = 1 Writes file
37	MINMR	500	Range to target within which program will automatically initiate process for miss distance computation
38	DTMIN	.001	Minimum integration step size for back-up to compute miss distance (program terminates if step is less than this value)
39	MCARLO	(1)	Flag for initiating Monte Carlo mode of operation = 0 No Monte Carlo = 1 Monte Carlo Mode
40	NTRYS	20	Number of executions for Monte Carlo mode
41	TIME	0	Running time (when entered as input data, it is equivalent to initial starting time, e.g., $t_o = 0$)
42	DTOP	.5	Print interval, i.e., time increment for printout
43	DT	.01	Initial (starting) value of numerical integration step size
44	TOTAL	30	Time limit placed on internal program running time if it is not terminated first by miss distance calculation

(1) Both 1 and 2 used.

Table A-II (Cont'd)

Data No.	Program Variable	Assigned Value	Description
45	IAUTO	1	Flag for defining autopilot transfer function or 6 DOF operation = 0 Calls LAG - first order exponential = 1 Calls APILOT second-order with lead lag terms = 2 Calls AUTOPL 6 DOF executive driver routine (not active at this time)
46	ACCTGT	9	Target acceleration magnitude (#g's) in maneuver
47	TGTROL	45	Target bank angle defining climbing/ diving turns in subroutine TURN3D
48	LPRINT	1	Print flag = 0 Allows all print = 1 Prints only miss distance = 2 Prints time and miss distance (Monte Carlo Mode)
49	JPRINT	1	Print Flag = 0 Detailed output (132 char.) = 1 Condensed output (132 char.) = 2 Condensed output (80 char.) for remote terminal
50	RNGTGO	100	Range-to-go before point of closest approach (set DATA (35) to large value if used)
51	IRAND	.85	Random number seed for operation in Monte Carlo code

Table A-II (Cont'd)

Data No.	Program Variable	Assigned Value	Description
52	ISTAT	1	Statistics Flag = 0 No statistics = 1 Fixed no. of trials. Calculate mean and standard deviation of miss distance and time of flight = 2 Fixed no. of trials. Calculate mean and standard deviation of miss distance and time of flight. Calculate percent error in mean miss distance estimate as function of confidence prob. (Significance Level) and no. of trials = 3 No. of trials determined sequentially based on required accuracy of mean miss dist. estimate and required statistical confidence
53	ISEEK	3	Seeker Flag = 0 No call ot generic seeker = 1 Call seeker thermal noise = 2 Call seeker glint noise = 3 Call seeker glint + thermal noise
54	TNOISE	$5 \cdot 10^{-10}$	Standard deviation for representing thermal noise
55	GNOISE	200	Standard deviation for representing glint noise
56	CONFID	.80001	Required statistical confidence probability
57	ERPCNT	30	Required percent error in Mean Miss Distance
58	THRST1	0	First stage rocket motor thrust (lbs)
59	THRST2	0	Second stage rocket motor thrust (lbs)
60	ALFAMX	21.8	Maximum missile angle of attack (degs)
61	IYY	94	Missile moment of inertia, pitch axis (slug-ft^2)

Table A-II (Cont'd)

Data No.	Program Variable	Assigned Value	Description
62	DELDMX	300	Missile maximum control surface deflection rate (degs/sec)
63	ROLLS	0	Number of rolls required for barrel roll routine
64	ROLLRT	0	Roll rate required for barrel roll routine (degs/sec)
65	PERIOD	0	Period defining jinking maneuver (secs)
66	WVLENG	1.2	Radar wave length (default = 1.2 inches)
67	APRTUR	10^6	Radar aperture (default = 10^6 inches)
68	FYNESS	2.4	Nose fineness ratio (default = 2.4)
69	FVAR	.02	Frequency variation (default = .02)
70		45	Seeker gimbal angle limit (deg). Default 45 deg.
71		60	Seeker gimbal rate limit, (deg/sec). Default 60 deg/sec.
72-79	(1)		
80	DELON	1	Flags that user wishes to execute revised Tactics IV
81	Delay1	(2)	Delay between seeker and guidance computer
82	Dealy2	(2)	Delay between guidance computer and autopilot
83	Delay3	(2)	Delay between inertial and guidance computer
84	Tcomp	(2)	Time into DT interval when pronav computation is completed

(1) Spare data values
(2) All of the delays vary between 0 and .01

Table A-II (Cont'd)

Data No.	Program Variable	Assigned Value	Description
85	Slagon	1	Flags that user wishes to employ the .1 second seeker lag
86-97	(1)		
98			Longitudinal net acceleration factor Default 1
99		3	Multiplier on ALFAMX (max angle of attack) to model parametrically the acceleration capability of the missile. Default 1
100	XREF		Static margin of missile (ft.). distance from CG to CP. Treated as a constant rather than a dynamic variable. Default 0.25 ft.

(1) Spare data values.

```
1
0.20    0.80    1.50    2.00    2.35    2.87    3.95    4.60

0.235   0.240   1.05    0.910   0.830   0.745   0.630   0.580

1.04    1.04    1.04    0.93    0.86    0.90    0.90    0.87

0.755   0.755   0.755   0.413   0.288   0.180   0.108   0.090

0.93    0.93    0.93    0.64    0.62    0.42    0.33    0.31

0.116   0.127   0.198   0.162   0.141   0.113   0.070   0.051

0.0     4.0     6.0     8.0     10.0    12.0    14.0    16.0    18.0
4.0     4.0     4.0     4.0     4.0     4.0     4.0     4.0     4.0
 CASE LFC: - HEADING ERROR,CLIMB-DIVE,STOCH,APDLAY=.0075

001 +0.0         +0.0        +20000.0     +2.50       +0.0

006 +0.00        +10000.0    +1000.0      +20000.0    +0.8

011 +00.0        +0.0        +1.0         +1.0        +1.0

016 +370.0       +1.0        +1.0         +0.0        +40.0

021 +0.0

022 +4.0         +0.0        +0.0         +0.0        +0.01

027 +0.01        +10.0       +.7          +100.       +0.0

032 +0.0         +.349066    +4.0         +1.0        +1.0

037 +500.0       +.0010      +1.0         +20.0

041 +0.0         +.5         +.01         +30.0       +1.0

046 +9.0         +45.0       +2.0         +1.0        +100

051 +0.85        +1.0        +3.0         +0.0        +0.0

058 +0.0         +0.0        +21.8        +94.0       +300.0

063 +0.0         +0.0        +0.0         +1.2        +0.0

068 +2.4         +0.02       +45.0        +60.0

080 +1.0         +.0000      +.00375      +.0000      +.00375

085 +1.0

099 +3.0
```

Figure A-1. Sample Tape5 File Listing

Appendix B

Modules Revised in Tactics IV

The Tactics IV program was altered for this study. The changes included both revising modules found in the original program and adding new modules to reflect the DIS missile model. This appendix contains the four revised modules. These modules are Apilot(I), Incond, Misilx(I), and and Pronav(I). This module listing includes both a descriptive header and the code itself. The lines of code which were modified or added to the original are designated by a "*" in the left margin. Additionally, only a portion of Incond was included due to the amount of code. Finally, the reader is referred to REF2 for a description and complete listing of original Tactics IV.

```
      SUBROUTINE APILOT(I)

**************************************************************************

*  CC  MODULE NAME:  APILOT(I)
*  CC  FUNCTION:  THIS ROUTINE CONVERTS COMMANDED ACCELERATION INTO
*  CC    OUTPUT ACCELERATION.  THE OUTPUT RESPONSE IS SIMPLE SECOND
*  CC    ORDER.  THIS REVISED TACTICSIV MODULE PERMITS
*  CC    THE USE OF DELAYS IN THE RECEIPT OF THE NEW COMMANDED
*  CC    ACCELERATION COMMAND.  IF THE NEW COMMAND IS DELAYED, THE
*  CC    LAST COMMAND ACCELERATION IS USED UNTIL IT ARRIVES.
*  CC  INPUTS:
*  CC    ACOM(I):       TOTAL COMMANDED ACCELERATION
*  CC    ACOMA(I):      HORIZONTAL COMMANDED ACCELERATION
*  CC    ACOMD(I):      VERTICAL COMMANDED ACCELERATION
*  CC    ACOMAD(I):     LAST COMMANDED HORIZONTAL ACCELERATION
*  CC    ACOMDD(I):     LAST COMMANDED VERTICAL ACCELERATION
*  CC    APDLAY:        AMOUNT ACOM IS DELAYED IN REFERENCE TO
*  CC                   THE BEGINNING OF THE INTEGRATION STEP DT
*  CC    DELT           SIZE OF TRAPEZOIDAL INTEGRATION STEP (SEC)
*  CC    HDP            SIZE OF CURRENT INTEGRATION STEP
*  CC    I              BODY IN QUESTION WHERE:
*  CC                     1=LAUNCH AIRCRAFT
*  CC                     2=MISSILE
*  CC                     3=TARGET
*  CC    NINT           NUMBER OF TRAPEZOIDAL INTEGRATION STEPS
*  CC                   CURRENT INTEGRATION STEP IS DIVIDED INTO
*  CC    OMEGAN         AUTOPILOT NATURAL FREQUENCY
*  CC    ZELTA          AUTOPILOT DAMPING RATIO
*  CC    TAU1 / TAU2    AUTOPILOT LEAD / LAG TERMS (1/1 WHEN DELON
*  CC                      IS SET TO ONE)
*  CC    TYME           CURRENT SIMULATION TIME
*  CC  OUTPUTS:
*  CC    AOUT(I):       TOTAL OUTPUT ACCELERATION
*  CC    AOUTA(I):      HORIZONTAL ACCELERATION
*  CC    AOUTD(I):      VERTICAL OUTPUT ACCELERATION
```

```
*   CC     PHIDOT(I):     TIME RATE OF CHANGE OF PHI
*   CC     THDOT(I):      TIME RATE OF CHANGE OF THETA
*   CC   MODULES CALLED:
*   CC     AUXAPD(EXTERNAL):  THIS MODULE EVALUATES DERRIVATIVES
*   CC       WHICH ARE USED BY THE TRAPEZOIDAL INTEGRATION ROUTINE.
*   CC       THE COMMANDED ACCELERATION USED IS ACOMD(I).
*   CC     AUXAPL(EXTERNAL):  THIS MODULE EVALUATES DERRIVATIVES
*   CC       WHICH ARE USED BY THE TRAPEZOIDAL INTEGRATION ROUTINE.
*   CC       THE COMMANDED ACCELERATION USED IS ACOM(I).
*   CC     TRAPZ:   THIS MODULE INTEGRATES THE AUTOPILOT RESPONSE
*   CC       EQUATIONS.
*   CC   CALLING MODULES:  B2TURN(I),MISILX(I)
*   CC
*   CC
*   CC
*   CC
*   CC
*   CCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
*   CC   INTEGER XDELT,CASEX
*   CCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
         REAL MACH,LAMBDA,LIFT,MINMR
         COMMON /CATMOS/DENS(3),SOUND(3),MACH(3),Q(3)
         COMMON /VECTOR/ R(3,6),RREL(3,6),V(3,6),VREL(3,6)
         COMMON /AERO/ THRUST(3),ALPHA(3),CODRAG(3),COLIFT(3),DRAG(3),
        1   LIFT(3),WEIGHT(3),AREA(3),ASMAX(3),CLMAX(3),WO(3)
         COMMON /RATE/ OMEGAR(3,4),OMEGAB(3,4),RDOT(3),VDOT(3),
        1   THDOT(3),PHIDOT(3)
         COMMON /ATT/ AZMUTH(6),ELEV(6),ROLL(3),ROLLR8(3),BEARING(6),ELEVMX(
        13),AZMAX(3),BANK(3)
         COMMON /ACCEL/ ACOM(3),ACOMA(3),ACOMD(3),AOUT(3),AOUTA(3),AOUTD(3)
        1,GFORCE(3),ACLMAX(3)
         COMMON /UNIT/ UNITV(3,3),UNITPV(3,3),UNITR(3,3),UNITL(3,3),UNITT(3
        1,3),A(3,3),D(3,3),A1(3,3),D1(3,3),UNITP(3,3),UNITY(3,3)
         COMMON /INTEG/ GDP,T,VAR(6),DER(6),TEMPS(2,6),VARSTR(6),
        1   STIME,TIME,HM,DT,DTIMIN,NINT
         COMMON /FLAG/ JPOL,KPOL,LPOL,MPOL,NPOL,ISTART,ILAUN,JATMOS,
```

```
     1   INIT8(2),MANUVR,IPOST
      COMMON/PRINT/ IPRINT(20),TOP,TOTAL,DTPO,NPRINT,TITLE(18),
     1   LPRINT,JPRINT
      COMMON/MISSL/TGUIDE,TLAUN,TBURN1,TBURN2,MINMR,WBURN,
     1   LABMDA,BRNR81,BRNR82,TGO,IAUTO,ACCTGT,TGTROL,RNGTGO,
     2   THRST1,THRST2,ALFAMX,IYY,DELDMX
      COMMON/ANALYT/CDO,SLOPE
      COMMON /CONST/ G,RAD,TAU(3),DATA(100),IZERO
      COMMON/EXTRA/ EXTR(20),IEXTRA(10)
      COMMON/VEH/ ACTION(3,2),DSMX(3,5),DUMCL(3,5),ACTNO(3),ASPECT
      COMMON/PARMS/OMEGAN,ZETA,TAU1,TAU2
      COMMON/AUTO/YVAR(6),YDER(6),YTEMPS(2,6),DUM(5)
*  CCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
*     COMMON/DELAY/ACOMAD(3),ACOMDD(3),DELON,DELAY1,DELAY2,
*    1DELAY3,TCOMP,APDLAY,DFLAG,LAGOUT,SAVEA(2,5),SLAGON
*
*  CCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
      EXTERNAL AUXAPL,AUXAPD

      KK=I-1
      IF(INIT8(KK) .EQ.1) GO TO 50
*  CC
*  CC   THE FOLLOWING CODE INITIALIZES CERTAIN VARIABLES CCCCCCCCCC
*  CC
      DUM(2)=1./TAU2
      DUM(3)=TAU1=DUM(2)
      TIME=0
      DUM(1)=(1.-DUM(3))=DUM(2)
      DO 25 J=1,6
   25 YVAR(J)=0.
      INT8(KK)=1
      DUM(4)=OMEGAN*OMEGAN
      DUM(5)=2.*ZETA*OMEGAN
*     ACOMAND(I)=ACOMA(I)
*     ACOMDD(I)=ACOMD(I)
*     CALL AUXAPD
```

```
      CALL AUXAPL
      RETURN
*  CC
*  CC   END OF INITIALIZATION CODE CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
*  CC
   50 CONTINUE
*  CC
*  CC   FOLLOWING CODE USED TO RUN OUT SEEKER LAG ONCE MISSILE
*  CC   GOES BLIND.
*  CC
*       IF (ISTART.NE.1)GO TO 55
*       DO 53 J=1,5
*       SAVEA(2,J)=SAVEA(1,J)
*    53 CONTINUE
*       SAVEA(1,1)=ACOM(2)
*       SAVEA(1,2)=ACOMA(2)
*       SAVEA(1,3)=ACOMD(2)
*       SAVEA(1,4)=ACOMAD(2)
*       SAVEA(1,5)=ACOMDD(2)
*    55 CONTINUE
*       IF(ISTART.EN.2)GO TO 58
*       ACOM(2)=SAVEA(2,1)
*       ACOMA(2)=SAVEA(2,2)
*       ACOMD(2)=SAVEA(2,3)
*       ACOMAD(2)=SAVEA(2,4)
*       ACOMDD(2)=SAVEA(2,5)
*  CC
*  CC
*    58 CONTINUE
        DUM(4)=OMEGAN*OMEGAN
        DUM(5)=2.*ZETA*OMEGAN
        IF(HM .LE. 0) GO TO 110
        DELT=HDP/NINT
*  CC
*  CC   _Y INTEGRATE THE ACOMD REGION IF REVISED TACTICSIV CCCCCCC
*  CC   IS SELECTED.                                       CCCCCCC
*  CC
```

```
*   CC   IF (DELON .EQ. 1) GO TO 60
*        XDELT=0
*        GO TO 95
*     60 CONTINUE
*   CC   ALSO, ONLY INTEGRATE THE ACOMD REGION IF THE MISSILE   CCCCC
*   CC   IS THE BODY IN QUESTION.                               CCCCC
*   CC
*         IF (I .EQ. 2) GO TO 65
*         XDELT=0
*         GO TO 95
*      65 CONTINUE
*   CC
*   CC   DETERMINE NUMBER OF TRAPEZOIDAL STEPS IN THE ACOMD REGION CC
*   CC
*         XDELT=INT(APDLAY/DELT + .5)
*   CC
*   CC   (DT/HDP) IS GREATER THAN ONE IN  THE LAST DT OF THE FIGHT CCC
*   CC
*         IF (DT/HDP .EQ. 1) GO TO 80
*         CALL LASTDT(CASEX,XDELT)
*         GO TO (80,95) CASEX
*      80 IF (XDELT .EQ. 0) GO TO 95
*
*   CC
*   CC   FOLLOWING  CODE INTEGRATES ACOMD REGION CCCCCCCCCCCCCCCCCCCCC
*   CC
*         DO 90 K=1,XDELT
*         CALL TRAPZ(TYME,DELT,YVAR,YDER,AUXAPD,6,YTEMPS)
*      90 CONTINUE
*   CC
*   CC    FOLLOWING CODE INTEGRATES ACOM REGION CCCCCCCCCCCCCCCCCCCCCCC
*   CC
*      95 XDELT=XDELT+1
*         IF (XDELT .EQ. NINT) GO TO 110
          DO 100 K=XDELT,NINT
          CALL TRAPZ(TYME,DELT,YVAR,YDER,AUXAPL,6,YTEMPS)
      100 CONTINUE
```

```
*   CC
*   CC   DEFINE THE OUTPUT VARIABLES CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
*   CC
      110 AOUTA(I)=DUM(3)*YVAR(2) + YVAR(3)
          AOUTD(I)=DUM(3)*YVAR(5) + YVAR(6)
          AOUT(I) = SQRT(AOUTA(I)**2 + AOUTD(I)**2)
          IF (DT/HDP .GT. 1) GO TO 120
*         ACOMAD(I)=ACOMA(I)
*         ACOMDD(I)=ACOMD(I)
      120 PHIDOT(I) = AOUTD(I)/V(I,4)
          IF (ABS(COS(V(I,6))) .LE. 1.0E-06) GO TO 250
          THDOT(I) = AOUTA(I)/(V(I,4)*COS(V(I,6)))
          GO TO 260
      250 THDOT(I) = 1.0E+08
      260 CONTINUE
          RETURN
          END
```

```
      SUBROUTINE INCOND
***************************************************************************
*   CC   MODULE NAME:  INCOND
*   CC
*   CC   FUNCTION:  THE ORIGINAL FUNCTION INITIALIZED KEY TACTICSIV
*   CC     VARIABLES AND INPUT THE TAPES DATA ARRAY.  THE REVISED FUNCTION
*   CC     DOES THIS ALONE WITH AN ADDITIONAL OPERATION.  THIS NEW OPERA-
*   CC     TION IS TO INITIALIZE SEVERAL NEW VARIABLES AND INPUT THOSE
*   CC     TAPES DATA ITEMS ASSOCIATED WITH REVISED TACTICSIV.  THIS
*   CC     NEW OPERATION IS ACCOMPLISHED IN THE DIGITAL SUBROUTINE.
*   CC   INPUTS (TO NEW CODE):
*   CC     DATA(80):               REVISED TACTICSIV FLAG (1=SELECTED)
*   CC   OUTPUTS(FROM NEW CODE):
*   CC     DELAY1                  SEEKER BUS DELAY
*   CC     DELAY2                  INITIAL BUS DELAY
*   CC     DELAY3                  GUIDANCE COMPUTER BUS DELAY
*   CC     TCOMP                   GUIDANCE COMPUTER COMPUTATION TIME
*   CC     APOLAY                  DELAYS PRECEDING THE AUTOPILOT
*   CC     DELON                   REVISED TACTICSIV FLAG (1=SELECTED)
*   CC     SLAGON                  SEEKER LAG FLAG (1=SELECTED)
*   CC   MODULES CALLED(BY NEW CODE):
*   CC     DIGITAL                 INITIALIZES REVISED TACTICS IV VARIABLES
*   CC   CALLING MODULES:
*   CC     MAIN
*   CC   NOTE:  THE ABOVE NARRATIVE DESCRIBES ONLY THE CODE ADDED
*   CC     ADDED IN REVISED TACTICS IV.
      REAL MACH,LAMBDA,LIFT,MINMR,IYY
      COMMON /CATMOS/ DENS(3),SOUND(3),MACH(3),Q(3)
      COMMON /VECTOR/ R(3,6),RREL(3,6),V(3,6),VREL(3,6)
      COMMON /AERO/ THRUST(3),ALPHA(3),CODRAG(3),COLIFT(3),DRAG(3),
     1  LIFT(3),WEIGHT(3),AREA(3),ASMAX(3),CLMAX(3),WO(3)
      COMMON /RATE/ OMEGAR(3,4),OMEGAB(3,4),RDOT(3),VDOT(3),
     1  THDOT(3),PHIDOT(3)
      COMMON /ATI/ AZMUTH(6),ELEV(6),ROLL(3),ROLLR8(3),BEARNG(6),ELEVMX(
```

```
      13),AZMAX(3),BANK(3)
       COMMON /ACCEL/ ACOM(3),ACOMA(3),ACOMD(3),AOUT(3),AOUTA(3),AOUTD(3)
      1,GFORCE(3),ACLMAX(3)
       COMMON /UNIT/ UNIT(3,3),UNITPV(3,3),UNITR(3,3),UNITL(3,3),UNITT(3
      1,3),!(3,3),D(3,3),A1(3,3),D1(3,3),UNITP(3,3),UNITY(3,3)
       COMMON /INTEG/ HDP,T,VAR(6),DER(6),TEMPS(2,6),VARSTR(6),
      1   STIME,TIME,HM,DT,DTMIN,NINT
       COMMON /FLAG/ JPOL,KPOL,LPOL,MPOL,NPOL,ISTART,ILAUN,JATMOS,
      1   INIT8(2),MANUVR,IPOST
       COMMON/PRINT/ IPRINT(20),TPO,TOTAL,DTPO,NPRINT,TITLE(18),
      1   LAMBDA,BRNR81,BRNR82,TGO,IAUTO,ACCTGT,TGTROL,RNGTOGO,
      2   THRST1,THRST2,ALFAMX,IYY,DELDMX
       COMMON/ANALYT/CDO,SLOPE
       COMMON /CONST/ G,RAD,TAU(3),DATA(100),IZERO
       COMMON /EXTRA/ EXTR(20),IEXTRA(10)
       COMMON/VEH/ ACTION(3,2),DASMX(3,5),DUMCL(3,5),ACTNO(3),ASPECT
       COMMON/PARAMS/OMEGAN,ZETA,TAU1,TAU2
       COMMON/MANVR/CV50,SV50,CV60,SV60
       COMMON/SIXDEG/ XDER(12),SVAR(12),XTEMPS(2,12)
       COMMON/TIMEX/AYB,AZB,MB,MC
       COMMON/INTEG1/VARIBL(18),DERIV(18),ZTEMPS(2,18)
       COMMON/ATTIT/XPP,XQQ,XRR,QDOT,RDOTX,ALFAT.BETAT,DR,DQ,DP
       COMMON/CARLO/ MCARLO,IRAND,NTRYS,ISTAT,KTRY,ISEEK,CONFID,
      1   ERPCNT,TNOISE,GNOISE
       COMMON/TARGT/ROLLS,ROLLRT,PERIOD
       COMMON/BORERR/WVLENG,APRTUR,FYNESS,FVAR
       COMMON/BLK3/DUM3(32)
       COMMON/BLK4/DUM4(8)
       COMMON/ACT/ACTST(3)
*      COMMON/DELAY/ACOMAD(3),ACOMDD(3),DELON,DELAY1,DELAY2,
*     1DELAY3,TCOMP,APDLAY,DFLAG,LAGOUT,SAVEA(2,5),SLAGON

       DATA BLANK/4H   /
    ********************************************************************

*****SKIP READING NEW DATA IF OPERATING MONTE CARLO MODE
```

```
          IF(DATA(39) .GT. 0.) GO TO 20
     **** READ DATA AND SET CONSTANTS

          DO 5 I = 1,18
        5 TITLE(I) = 0.0
          READ(5,750) (TITLE(J),J=1,18)
          IF(EOF(5)) 999,10
       10 CALL DECRD (DATA)

          IRAND=DATA(51)
          KTRY=0
       20 CONTINUE

     **** SET INITIAL FLAGS AND ZERO OUT VARIABLES

          IF(KTRY .GT. 0) IRAND=IRAND+2
          IF(IRAND .GT. 0) CALL RANSET(IRAND)

*    CC
*    CC   THIS CODE ADDED FOR REVISED TACTICSIV CCCCCCCCC
*    CC
*
*         IF (DATA(50).NE.1) GO TO 30
*         CALL DIGITL
*         GO TO 35

*    CC
*    CC   IF DELON IS NOT 1, SET ALL THE DELAYS TO ZERO CCC
*    CC
*      30 CONTINUE
*         DELAY1=0
*         DELAY2=0
*         DELAY3=0
*         TCOMP=0
*         APDLAY=0
```

```
*       DELON=0
*       SLAGON=0
*    35 CONTINUE
```

```
      SUBROUTINE MISILX(I)
************************************************************************
*   CC   MODULE NAME:  MISILX
*   CC   FUNCTION:  THIS ROUTINE IS THE MASTER MISSILE MODULE.
*   CC     ALL MODULE FUNCTIONS INCLUDING ROCKET BURN, CALCULATION
*   CC     OF AERODYNAMICS, SEEKER COMPUTATIONS, ETC. ARE CALLED
*   CC     BY THIS MDOULE.  ADDITIONAL CODE WAS ADDED FOR
*   CC     TACTICSIV.  THIS CODE PERMITS THE RUNOUT OF THE .1
*   CC     SECOND SEEKER LAG SHOULD THE MISSILE GO BLIND.
*   CC   INPUTS(TO NEW CODE):
*   CC     ISTART                 ENGAGEMENT PHASE FLAG (2=TERMINAL)
*   CC     DELON                  REVISED TACTICSIV FLAG (1=SELECTED)
*   CC     IEXTRA(10)             SEEKER LOCK FLAG (0=LOCKED)
*   CC     DATA(80)               REVISED TACTICSIV FLAG (1=SELECTED)
*   CC     LAGOUT                 NUMBER OF ITERATION CYCLES
*   CC   OUTPUTS(FROM NEW CODE):  SEEKER HAS BEEN BLIND
*   CC     NONE.
*   CC   MODULES CALLED(BY NEW CODE):
*   CC     NONE.
*   CC   CALLING MODULES
*   CC     POLICY.
*   CC   NOTE:  THE ABOVE NARRATIVE EXPLAINS ONLY CODE
*   CC          ADDED IN REVISED TACTICSIV.
      REAL MACH,LAMBDA,LIFT,MINMR,IYY,LAMTAB,MACHTB
      COMMON /CATMOS/ DENS(3),SOUND(4),MACH(3),G(3)
      COMMON /VECTOR/ R(3,6),RREL(3,6),V(3,6),VREL(3,6)
      COMMON /AERO/ THRUST(3),ALPHA(3),CODRAG(3),COLIFT(3),DRAG(3),
     1   LIFT(3),WEIGHT(3),AREA(3),ASMAX(3),CLMAX(3),WO(3)
      COMMON /RATE/ OMEGAR(3,4),OMEGAB(3,4),RDOT(3),VDOT(3),
     1 THDOT(3),PHIDOT(3)
      COMMON /ATT/ AZMUTH(6),ROLL(3),ROLLR8(3),BEARING(6),ELEVMX(
     13),AZMAX(3),BAND(3)
      COMMON /ACCEL/ ACOM(3),ACOMA(3),ACOMD(3),AOUT(3),AOUTA(3),AOUTD(3)
     1,GFORCE(3),ACLMAX(3)
```

```
      COMMON /UNIT/ UNITV(3,3),UNITPV(3,3),UNITR(3,3),UNITL(3,3),UNIT(3
     1,3),A(3,3),D(3,3),A1(3,3),D1(3,3),UNITP(3,3),UNITY(3,3)
      COMMON /INTEG/ HOP,T,VAR(6),DER(6),TEMPS(2,6),VARSTR(6),
     1   STIME,TIME,HM,DT,DTMIN,NINT
      COMMON /FLAG/ JPOL,KPOL,LPOL,MPOL,NPOL,ISTART,ILAUN,JATMOS,
     1   INIT8(2),MANUVR,IPOST
      COMMON/PRINT/ IPRINT(20),TPO,TOTAL,DTPO,NPRINT,TITLE(18),
     1   LPRINT,JPRINT
      COMMON/MISSL/TGUIDE,TLAUN,TBURN1,TBURN2,MINMR,WBURN,
     1   LAMBDA,BRNR81,BRNR82,TGO,IAUTO,ACCTGT,TGTROL,RNGTGO,
     2   THRST1,THRST2,ALFAMX,IYY,DELDMX
      COMMON/ANALYT/CDO,SLOPE
      COMMON /CONST/ G,RAD,TAU(3),DATA(100),IZERO
      COMMON/EXTRA/ EXTR(20),IEXTRA(10)
      COMMON/VEH/ ACTION(3,2),DASMX(3,5),DUMCL(3,5),ACTNO(3),ASPECT
      COMMON/TAGLE/JVEH,MACHTB(15),CDOTAB(15),CNATAB(15),CMDTAB(15),
     *   ADTAB(15),DELCDT(15),TYMTAB(10),LAMTAB(10)
      COMMON /PARAMS/ OMEGAN,ZETA,TAU1,TAU2
      COMMON/BLK9/APP(64)
      COMMON/DELAY/ACOMAD(3),ACOMDD(3),DELON,DELAY1,DELAY2,
     1DELAY3,TCOMP,APDLAY,DFLAG,LAGOUT,SAVEA(2,5),SLAGON

      EQUIVALENCE(T,XDP)
      DIMENSION AMISS2(2)
************************************************************************

      DATA AMISS2/4HMISI,4HLX  /
      DO 50 J=1,2
   50 ACTION(I,J)=AMISS2(J)
 **** SET CONSTANTS

      TIME=XDP
      IF(ACTNO(I).EQ.22.0) GO TO 75
      TBURN=TBURN1+TBURN2
```

```
 FIRST STAGE BURNOUT WEIGHT
    WO1=WC(I)+BRNR81*TBURN1

 SECOND STAGE BURNOUT WEIGHT
    WO2=WO1+BRNR82*TBURN2
    DIAM=SQRT(4.0 * AREA(2)/3.1415927)
    DUMY=2.15*SQRT(ZETA)
    THRUST(I)=THRST1
    ITH=0
    INITB(1)=0
    CALL ATITUD(I)
    XMACH=1.8
    CALL TABINT(XMACH,DCNDA,MACHTB,CNATAB,15)
    CN=DCNDA*ALFAMX/2.
    ALFAMX=ALFAMX*RAD*DATA(99)
    COSALF=COS(ALFAMX)
    SINALF=SIN(ALFAMX)
    XREF=DATA(100)
 75 CONTINUE

  END INITIALIZATION

    CALL ATMOS(I)
    IF(TIME .GE. TBURN1) GO TO 100
    THRUST(I)=THRST1
    WEIGHT(I)=WO(I) + BRNR81*TIME
    GO TO 130
100 CONTINUE
    IF(TIME .GE. TBURN) GO TO 125
    WEIGHT(I)=WO1+BRNR82*(TIME-TBURN1)
    THRUST(I)=THRST2
    GO TO 130
125 CONTINUE
    ITH=1
    THRUST(I)=0.
130 CONTINUE
```

```
 CALCULATE CLMAX BASED ON ALFAMAX (UNITS IN DEGS)

      CALL TABINT(MACH(I),OCNDA,MACHTB,CNATAB,15)
      CNMAX=DCNDA*ALFAMX/RAD
      CALL TABINT(MACH(I),CDO,MACHTB,CDOTAB,15)
      IF(ITH .EQ. 0) GO TO 140

 ADJUST CDO FOR MOTOR BURNOUT

      CALL TABINT(MACH(I),DELCD,MACHTB,DELCDT,15)
      CDO=CDO+DELCD
 140  CONTINUE
      CAMAX=CDO/COSALF
      CLMAX(I)=CNMAX*COSALF-CAMAX*SINALF
      SLOPE=CLMAX(I)/ALFAMX
      AQ=AREA(I)*Q(I)
 *******THE FOLLOWING COMPUTATION OF TIME CONSTANT TAU IS OPTIONAL
        DEPENDING ON WHETHER TAU(2) IS ENTERED AS DATA VALUE (DATA(24

      QUS=AQ*DIAM
      IF((DATA(24).GT.0.) .OR. (DATA(28).FT.0.)) GO TO 143
 ***************CHANGES BELOW TO REPRESENT SECOND ORDER RESPONSE******
  SET XREF USING DATA(100)--TYPICAL VALUE FOR .25 EQUIV. 1ST ORDER TIM
 SET IAUTO -DATA($%)- TO VALUE OF UNITY FOR SECOND ORDER TRANSFER FUNC
      CALL TABINT(MACH(I),DCNDA,MACHTB,CNATAB,15)
      CN=DCNDA*ALPHA(I)/RAD
      CNALFA=DCNDA/RAD
      CMALFA=CNALFA*XREF/DIAM
      OMEGAN=SQRT(CMALFA*QSD/IYY)

********************ABOVE ADDED BY JHH TO CALC. NATURAL FREQUENCY******
 143  CONTINUE
      IF(DATA(20) .EQ. 0.) ASMAX(I)=G*(CLMAX(I)*AQ +THRUST(I)*
     1   SINALF)/WEIGHT(I)
```

```
      USING PROPORTIONAL NAVIGATION DEFINE COMMANDED ACCELERATION VECTOR
                ACOMA=HORIZONTAL PLANE ACCELERATION COMPONENT
                ACOMD=VERTICAL PLANE ACCELERATION COMPONENT
*        IF(ISTART.EQ.2.AND.DELON.EQ.1)GO TO 147
*   CC
*   CC    IF SEEKER BLIND AND REVISED TACTICSIV NOT CALLED  CCCC
*   CC    THEN SKIP PRONAV(I) AND USE LAST COMMANDED ACCEL- CCCC
*   CC    ERATION TO  DRIVE APILOT(I).                      CCCC
*   CC
*           IF(IEXTRA(10) .GT. 0 .AND. DATA(80) .NE. 1) GO TO 147
*
*   CC    IF SEEKER NOT BLIND AND EITHER REVISED TACTICSIV NOT  C
*   CC    CALLED OR REVISED TACTICSIV NEVER HAS HAD A SEEKER CCC
*   CC    BREAK LOCK, CALL PRONAV(I) NORMALLY.               CCC
*   CC
*   CC
*         IF(IEXTRA(10) .EQ. 0 .AND.LAGOUT .EQ. 0)GO TO 145
*         IF(LAGOUT .GT. 10) GO TO 147
*         LAGOUT=LAGOUT+1
    145 CONTINUE
        CALL PRONAV(I)
    147 CONTINUE

      LIMIT COMMANDED VALUES TO MINIMUM OF ASMAX/CLMAX LIMITS

        CALL LIMIT(I)
    149 CONTINUE

    ********REPRESENT FIRST ORDER EXPONENTIAL TIME LAG RESPONSE

        IF(IAUTO .EQ. 0) CALL LAG(I)
      REPRESENT SECOND ORDER TRANSFER FUNCTION WITH LEAD/LAG
        IF(IAUTO .EQ. 1) CALL APILOT(I)
   **** COMPUTE AERODYANICS

        GFORCE(I)=SORT((AOUTA(I)/G)**2+(AOUTD(I)/G+COS(V(I,6)))**2)
```

```
 COMPUTE BALANCE OF AERO AND GRAVITATIONAL FORCES
  SOLVE TRANSEDENTAL LIFT + THRUST*SIN(ALPHA) EQUATION FOR ALPHA

--NEWTON RAPHSON - ALPHA INDEPENDENT VARIABLE
     FUNC1=100.
     DEL=.020
     IF(LAGOUT .EQ. 0) GO TO 155
 155 CONTINUE
     DO 200 J=1,10
     CALL TABINT(MACH(I),DCNDA,MACHTB,CNATAB,15)
     CN=DCNDA*ALPHA(I)/RAD
     CALL TABINT(MACH(I),CDO,MACHTB,CDOTAB,15)
     CN=DCNDA*ALPHA(I),CDO,MACHTB,CDOTAB,15)
     CALL TABINT(MACH(I),CDO,MACHTB,CDOTAB,15)
     IF(ITH .EQ. 0) GO TO 160
     CALL TABINT(MACH(I),DELCD,MACHTB,DELCDT,15)
     CDO=CDO+DELCD
 160 CONTINUE
     CA=CDO/COS(ALPHA(I))
     COLIFT(I)=CN*COS(ALPHA(I))-CA*SIN(ALPHA(I))
     FUNC2=GFORCE(I)*WEIGHT(I)-COLIFT(I)*AQ-THRUST(I)*
    1  SIN(ALPHA(I))
     IF(ABS(FUNC2).LT. 1.E-2)GO TO 210
     IF(J.EQ.1)GO TO 180
     DEL=DEL*FUNC2/(FUNC1-FUNC2)
 180 CONTINUE
     FUNC1=FUNC2
--INCREMENT ALPHA TO OBTAIN NEW CN,CA,CL
     ALPHA(I)=ALPHA(I)+DEL
 200 CONTINUE
 210 CONTINUE
     LIFT(I)=COLIFT(I)*AQ
     CODRAG(I)=CA*COS(ALPHA(I))+CN*SIN(ALPHA(I))
     DRAG(I)=CODRAG(I)*AQ
     LIFT(I)=COLIFT(I)*AREA(I)*Q(I)
VDOT(I)=-G*SIN(V(I,6))+G*THRUST(I)*COS(ALPHA(I))/WEIGHT(I)
```

```
     1 -G*DRAG(I)/WEIGHT(I)
       CALL ATITUD(I)
       IF(JPRINT .NE. 0) GO TO 450
       THDOT(2)=AOUTA(2)/(V(2,4)*COS(V(2,6)))
       PHIDOT(2)=AOUTD(2)/V(2,4)
   450 CONTINUE
       ACTNO(I)=22.
   500 CONTINUE
       RETURN
       END
```

```
      SUBROUTINE PRONAV(I)
      PROPORTIONAL NAVIGATION GUIDANCE LAW (GAIN=LAMBDA*VM)

***************************************************************************

*  CC   MODULE NAME:  PRONAV(I)
*  CC   FUNCTION:  THE ORIEINAL ROUTINE CALLS THE SEEKER MODULES AND
*  CC     PERFORMS THE PROPORTIONAL NAVIGATION COMPUTATIONS.  THE CODE
*  CC     ADDED FOR A REVISED THETICS IV INJECTS THE SEEKER LAG,
*  CC     INJECTS THE A10 CONVERTER NOISE AND SETS THE BUS DELAY
*  CC     ASSOCIATION WITH THE GUIDANCE COMPUTER.
*  CC   INPUTS (TO NEW CODE):
*  CC     DECON:  REVISED TACTICS IV FLAG (1=SELECTED)
*  CC     SLAGON:  SEEKER LAG FLAG (1=SELECTED)
*  CC     DATA(34):  MONTE CARLO MODE FLAG (1=SELECTED)
*  CC     RREL(3,4):  RANGE TO TARGET
*  CC     EXTR(2):  SEEKER BLIND RANGE
*  CC     IEXTRA(10):  BREAK LOCK STATUS (NONZERO IMPLIES LOCK BROKEN)
*  CC     LAGOUT:  NUMBER OF CYCLES LOCK HAS BEEN BROKEN
*  CC   OUTPUTS:
*  CC     NONE
*  CC   MODULES CALLED:
*  CC     SEKLAG:  -1 SECOND SEEKER LAG
*  CC     ADNOIS:  A10 CONVERSION NOISE
*  CC     BUS DLY:  BUS DELAYS PRIOR TO GUIDANCE COMPUTER
*  CC   CALLING MODULES:
*  CC     MISILX
*  CC   NOTE:  THE ABOVE NARRATIVE DESCRIBES ONLY THE CODE ADDED TO REQUIRED TACTICSIV.
        REAL MACH,LAMBDA,LIFT,MINMR,LAMTAB,MACHTB
        COMMON /CATMOS/ DENS(3),SOUND(3),MACH(3),Q(3)
        COMMON /VECTOR/ R(3,6),RREL(3,6),V(3,6),VREL(3,6)
        COMMON /AERO/ THRUST(3),ALPHA(3),CODRAG(3),COLIFT(3),DRAG(3),
       1  LIFT(3),WEIGHT(3),AREA(3),ASMAX(3),CLMAX(3),WO(3)
        COMMON /RATE/ OMEGAR(3,4),OMEGAB(3,4),RDOT(3),VDOT(3),
       1  THDOT(3),PHIDOT(3)
        COMMON /ATT/ AZMUTH(6),ELEV(6),ROOL(3),ROLLR8(3),BEARING(6),ELEVMX(13),AZMAX(3),BANK(3)
        COMMON /ACCEL/ ACOM(3),ACOMA(3),ACOMD(3),AOUT(3),AOUTA(3),AOUTD(3)
       1,GFORCE(3),ACLMAX(3)
```

```
      COMMON /UNIT/ UNITV(3,3),UNITPV(3,3),UNITR(3,3),UNITL(3,3),UNITT(31,3),A(3,3),D(3,3),
     UNITP(3,3)UNITY(3,3)
      COMMON /INTEG/ HDP,T,VAR(6),DER(6),TEMPS(2,6),VRSTR(6),
     1   STIME,TIME,HM,DT,DTMIN,NINT
      COMMON /FLAG/JPOL,KPOL,LOPL,MPOL,NPOL,ISTART,ILAUN,JATMOS,
     1   INIT8(2),MANUVAR,IPOST
      COMMON/PRINT/ IPRINT(20),TPO,TOTAL,DTPC,NPRINT,TITLE(18),
     1   PRINT,JPRINT
      COMMON/MISSL/TGUIDE,TLAUN,TBURN1,TBURN2,MINMR,WBURN,
     1   LAMBDA,BRNR81,BRNR82,TGO,IAUTO,ACCTGT,TGROL,RNGTGO,
     2   THRST1,THRST2,ALFAMX,IYY,DELDMX
      COMMON/ANALYT/CDO,SLOPE
      COMMON /CONST/ G,RAD,TAU(3),DATA(100),IZERO
      COMMON/EXTRA/ EXTR(20),IXTRA(10)
      COMMON/VEH/ ACTION(3,2),DASMX(3,5),DUMCL(3,5),ACTNO(3),ASPECT
      COMMON/TABLE/JVEH,MACHTB(15),CDOTAB(15),CNATB(15),CMDTAB(15),
     *   ADTAB(15),DELCDT(15),TYMTAB(10),LAMTAB(10)
      COMMON/CARLO/ MCARLO,IRAND,NTRYS,ISTAT,KTRY,ISEEK,CONFID,
     1   ERPCNT,INOISE,GNOISE
      COMMON/RSTOR/RRELST(6)
CCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
      COMMON/DELAY/ACOMAD(3),ACOMDD(3),DELON,DELAY1,DELAY2,
     1DELAY3,TCOMP,APDLAY,DFLAG,LAGOUT,SAVEA(2,5),SLAGON

CCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
      DIMENSION DUMV(3,3)CR(3),LASTWR(4)
      DATA DUM/0.0/

      IF(ISEEK .EQ. 0) TO TO 60
******STORE UNCORRUPTED VALUES OF RELATIVE RANGE VECTOR

      DO 25 J=1,6
   25 RRELST(J)=RREL(3,J)
****INITIALIZE BORESIGHT ERROR SUBROUTINE
      IF(ISTART .EQ. 0) JSTART=0
```

```
            IF(ISTART .EQ. 0) CALL BSERR(JSTART,DUM,DUM,DUM)
            CALL SEEKER
            CALL RATES
      ******RESTORE UNCORRUPTED VALUES OF RELATIVE RANGE VECTOR
            DO 50 J=1,6
         50 RREL(3,J)=RRELST(J)
         60 CONTINUE
            IF(ISEEK .EQ. 0) CALL RATES
            IF(OMEGAR(3,4).LT.1.E-8)OMEGAR93,4)=1.DE-4
*           IF (DELON .NE. 1) GO TO 85
*        80 IF (SLAGON.EQ.1)CALL SEKLAG
*     CC
*     CC  IF REVISED TACTICSIV STOCHASTIC NOISES ENGAGED,
*     CC  INJECT ANALOG-DIGITAL CONVERTER ERROR.
*     CC
*           IF (DATA(39).EQ.1.NAD.DELON.EQ.1)CALL ADNOIS
         85 CONTINUE
            DO 90 J=1,3
         90 DUMV(1,J)=OMEGAR(3,J)/OMEGAR(3,4)

         IF LAMBDA(DATA(22)) IS SET TO NONZERO VALUE, BYPASS TABULAR VALUES.
            IF (DATA(22) .GT. 0.) TO TO 95
            CALL TABINT(TIME,LAMBDA,TYMTAB,LAMTAB,10)
         95 CONTINUE

      --PROPORTIONAL NAVIGATION

*           IF(RREL(3,4) .LT. EXTR(2) .AND. DELON .NE. 1) GOTO 950
*           IF(IXTRA(10) .NE. 0 .AND. DELON .NE. 1) GO TO 950
*           IF(DELON .NE. 1) TO TO 100
*           IF(LAGOUT .GT. 10) GO TO 950
*           IF(LAGOUT .GE. 11) GO TO 950
*     CC
*     CC    ONLY CALL BUSDLY FOR REVISED TACTICSIV     CCCCCCC
*     CC
*           CALL BUSDLY
*     CC    THIS CODE COMPUTES COMMANDED ACCELERATION  CCCCCCC
*     CC
        100 CONTINUE
```

```
*     CC   THIS CODE COMPUTES COMMANDED ACCELERATION   CCCCCCC
 *    CC
      100 CONTINUE

          ACOM(I)=LAMBDA*V(2,4)*OMEGAR(3,4)
 *        IF(DATA(39).EQ.1.AND.DELON.EQ.1)CALL GCNOIS
          CR(1)=DUMV(1,2)*UNITV(2.3)-UNITV(2,2)*DUMB(1,3)
          CR(2)=-DUMV(1,1)*UNITV(2,3)+UNITV(2,1)*DUMV(1,3)
          CR(3)=DUMV(1,1)*UNITV(2,2)-UNITV(2,1)*DUMV(1,2)
          DUMX=SQRT(CR(1)**2+CR(2)**2+CR(3)**2)
          IF(DUMX .LT. 1.E-4) GO TO 110
          DO 101 J=1,3
      101 UNITPV(I,J)=CR(J)/DUMX
          GO TO 120
      110 CONTINUE
          UNITPV(I,1)=0.
          UNITPV(I,2)=0.
          UNITPV(I,3)-1.
      120 CONTINUE
          DOT1=UNITPV(T,1)*A(I,1)+UNITPV(I,2)*A(I,2)
          ACOMA(I)=ACOM(I)*DOT1
          DOT2=UNITPV(T,1)ID(I,1)+UNITPV(I,2)ID(I,2)+UNITPV(I,3)*
         1   D(I,3)
          ACOMD(I)=ACOM(I)*DOT2
      950 CONTINUE

      *** SET IEXTR(10) FLAG IF SEEKER IS BLIND AND LOCK-ON WASN'T BROKEN
          IF((IEXTRA(10).EQ.0) .AND. (RREL(3,4).LT.EXTR(2))) GOTO 998
          RSAVE=RREL(3,4)
          EXTR(4)=RSAVE

          RETURN
      998 CONTINUE
          IEXTRA(10)=3
          IF(IEXTRA(8).EQ.1) GOTO 999
```

```
      RATIOB=(EXTR(2)-RREL(3,4))/(RSAVE-RREL(3,34))
      EXTR(3)=T1RATIOB*HM
      IEXTRA(8)=1
  999 CONTINUE
      RETURN
      END
```

Appendix C

New Tactics IV Modules

The Tactic IV computer program was modified for this study. This modification included both altering existing modules (Appendix B) and adding new modules (Appendix C). Altogether, seven modules were created. These modules are Adnois, Auxapd, Busdly, Digitl, Gcnois, Lastdt and Seklag. The modules appear in alphabetical order along with a descriptive header that explains the code.

```
CC****************************************
      SUBROUTINE ADNOIS
CC****************************************
CC
CC    MODULE NAME:  ADNOIS
CC    FUNCTION:  THIS ROUTINE INJECTS A NOISE
CC      INTO THE LINE OF SIGHT RATE.  THIS
CC      NOISE COMES FROM THE ANALOG TO DIGITAL
CC      CONVERSION OF THE LINE OF SIGHT RATE.
CC      THIS NOISE IS ZERO MEAN AND IS NORMALLY
CC      DISTRIBUTED.  THE STANDARD DEVIATION IS
CC      DETERMINED BY BOTH THE NUMBER OF BITS
CC      OF THE CONVERTER AND THE SIGNAL RANGE
CC      THE CONVERTER CAN EXPECT.
CC    INPUTS:
CC      OMEGAR(3,J):        LINE OF SIGHT RATE
CC    OUTPUTS:
CC      OMEGAR(3,J):        LINE OF SIGHT RATE
CC    MODULES CALLED:
CC      ANRMRN:             NORMAL DISTRIBUTION
CC                          GENERATION FUNCTION
CC    CALLING MODULES:
CC      PRONAV(I)
CC
CC
      REAL ADSIG,ADERR,K,M,N,G
      COMMON/RATE/ OMEGAR(3,4),OMEGAB(3,4),RDOT(3),VDOT(3),
     1 THDOT(3),PHIDOT(3)

CC    COMPUTE THE STANDARD DEVIATION OF THE NOISE

       M=1
       N=16
       Q=M/(2**N)
       K=12
       ADSIG=Q/SQRT(K)
```

```
CC   INJECT CONVERSION ERROR INTO THE GHREE COMPONENTS
CC   OF LOS RATE SEPERATELY,
     DO 250 J=1,3
     ADERR=ANRMRN(ADSIG,0)
     IF(ADERR,GT.0)GO TO 50
     IF(ABS(ADERR).GT.(SQRT(K)*ADSIG))
    1ADERR=(-1)*(Q/2)
     GO TO 100
  50 IF(ABS(ADERR).GT.(SQRT(K)*ADSIG))
    1ADERR=Q/2
 100 OMEGAR(e,J)=OMEGAR(3,J)+ADERR
 250 CONTINUE
     OMEGAR(3,4)=SQRT(OMEGAR(3,1)**2+OMEGAR(3,2)**2+
    1OMEGAR(3,3)**2)
     RETURN
     END
```

```
*****************************************
     SUBROUTINE AUXAPD
*****************************************
CC   MODULE NAME:  AUXAPD
CC   FUNCTION:  THIS ROUTINE COMPUTES DERRIVATIVES TO BE USED BY THE
CC     TRAPEZOIDAL INTEGRATION ROUTINE.  THE COMMANDED ACCELERATION
CC     USED IS THE LAST COMMANDED ACCELERATION, ACOMAD AND ACOMDD.
CC   INPUTS:
CC     ACOMAD(I):        LAST HORIZONTAL COMMANDED ACCELERATION
CC     ACOMDD(I):        LAST VERTICAL COMMANDED ACCELERATION
CC     DUM(5)ARRAY:      DESCRIBES THE 2ND ORDER RESPONSE OF APILOT
CC     YVAR(6) ARRAY:    ACCELERATION COMPUTED BY TRAPEZOIDAL
CC                       INTEGRATION ROUTINE (TRAPZ)
CC   OUTPUTS:
CC     YDER(5) ARRAY:    ACCELERATION DERRIVATIVES FOR TRAPZ
CC   MODULES CALLED:  NONE
CC   CALLING MODULES:  APILOT(I),TRAPZ
CC
CC
CC
CC
      REAL MACH,LAMBDA,LIFT,MINMR
      COMMON /CATMOS/ DENS(3),SOUND(3),MACH(3),Q(3)
      COMMON /VECTOR/ R(3,6),RREL(3,6),V(3,6)VREL(3,6)
      COMMON /AERO/ THRUST(3),ALPHA(3),CODRAG(3),COLIFT(3),DRAG(3),
     1   LIFT(3),WEIGHT(3),AREA(3),ASMAX(3),CLMAX(3),WO(3)
      COMMON /RATE/ OMEGAR(3,4),OMEGAB(3,4),RDOT(3),VDOT(3),
     1  THDOT(3),PHIDOT(3)
      COMMON /ATT/ AZMUTH(6),ELEV(6),ROLL(3),ROLLRB(3),BEARNG(6),ELEVMX(
     13),AZMAX(3),BANK(3)
      COMMON /ACCEL/ ACOM(3),ACOMA(3),ACOMD(3),AOUT(3),AOUTA(3),AOUTD(3)
     1,GFORCE(3),ACLMAX(3)
      COMMON /UNIT/ UNITV(3,3),UNITPV(3,3),UNITR(3,3),UNITL(3,3),UNITT(3
     1,3),A(3,3),D(3,3),A1(3,3),D1(3,3),UNITP(3,3),UNITY(3,3)
      COMMON /INTEG/ HDP,T,VAR(6),DER(6),TEMPS(2,6),VARSTR(6),
```

```
1  STIME,TIME,HM,DT,DTMIN,NINT
 COMMON /FLAG/ JPOL,KPOL,LPOL,MPOL,NPOL,ISTART,ILAUN,JATMOS,
1    INIT8(2),MANUVR,IPOST
 COMMON/PRINT/ IPRINT(20),TPO,TOTAL,DTPO,NPRINT,TITLE(18),
1    LAMBDA,BRNR81,BRNR82,TGO,IAUTO,ACCTGT,TGTROL,RNGTGO,
2    THRST1,THRST2,ALFAMX,IYY,DELOMX
 COMMON/ANALYT/CDO,SLOPE
 COMMON /CONST/ G,RAD,TAU(3),DATA(100),IZERO
 COMMON/EXTRA/ EXTR(20),IEXTRA(10)
 COMMON/VEH/ ACTION(3,2),DASMX(3,5),DUMCL(3,5),ACTNO(3),ASPECT
 COMMON/AUTO/YVAR(6),YDER(6),YTEMPS(2,6),DUM(5)
 COMMON/DELAY/ACOMAD(3),ACOMDD(3),DELON,DELAY1,DELAY2,
1DELAY3,TCOMP,APDLAY,DFLAG,LAGOUT,SAVEA(2,5),SLAGON

 YDER(1)= -DUM(4)*YVAR(2)+DUM(4)*ACOMAD(2)
 YDER(2)= YVAR(1)-DUM(5)*YVAR(2)
 YDER(3)= DUM(1)*YVAR(2)-DUM(2)*YVAR(3)
 YDER(4)= -DUM(4)*YVAR(5)+DUM(4)*ACOMDD(2)
 YDER(5)= YVAR(4)-DUM(5)*YVAR(5)
 YDER(6)= DUM(1)*YVAR(5)-DUM(2)*YVAR(6)
 RETURN
```

```
YDER(6)= DUM(1)*YVAR(5) - DUM(2)*YVAR(6)
RETURN
END
```

```
****************************************
     SUBROUTINE BUSDLY
****************************************
CC   MODULE NAME:  BUSDLY
CC   FUNCTION:  THIS ROUTINE DETERMINES WHAT PARAMETERS WILL BE USED
CC     IN THE PRONAV COMPUTATION.  THAT IS, IT DETERMINES WHAT LOS
CC     RATE AND MISSILE VELOCITY DATA WILL HAVE ARRIVED AT THE
CC     GUIDANCE COMPUTER BY COMPUTE TIME.  IF THE NEWEST LOS RATE
CC     HAS ARRIVED, IT IS USED, IF NOT, THE LAST RECEIVED LOST RATE
CC     MUST BE USED.  SIMILARLY, THE NEW MISSILE VELOCITY DATA WILL
CC     BE USED IF IT HAS ARRIVED.  IF NOT, THE LAST RECEIVED
CC     MISSILE VELOCITY WILL BE USED.
CC   INPUTS:
CC     DELAY1:              TIME THAT LOS RATE IS DELAYED BETWEEN THE
CC                          SEEKER COMPUTER AND THE GUIDANCE COMPUTER.
CC                          THIS TIME IS IN REFERENCE TO THE BEGIN-
CC                          NING OF THE CURRENT DT INTEGRATION INTER-
CC                          VAL.
CC     DELAY3:              TIME THAT MISSILE VELOCITY IS DELAYED
CC                          BETWEEN THE INERTIAL REFERENCE COMPUTER
CC                          AND THE GUIDANCE COMPUTER.
CC                          THIS TIME IS IN REFERENCE TO THE BEGIN-
CC                          NING OF THE CURRENT DT INTEGRATION INTER-
CC                          VAL.
CC     ISTART:              START FLAG IN THE MAIN PROGRAM LOOP OF
CC                          TACTICSIV.
CC     OMEGAR93,4) ARRAY:   LOS RATE
CC     TCOMP:               TIME GUIDANCE COMPUTER CALCULATES PRONAV
CC                          FORMULA.  THIS TIME IS IN REFERENCE TO THE
CC                          BEGINNING OF THE CURRENT OT INTEGRATION
CC                          INTERVAL.
CC     UNITV(3,3) ARRAY:    MISSILE VELOCITY DIRECTION
CC     V(2,4):              MISSILE VELOCITY MAGNITUDE
CC   OUTPUTS:
CC     DUMV(3,3) ARRAY:     LOS RATE IN SPECIAL FORMAT FOR PRONAV
CC     OMEGAR(3,4) ARRAY:   LOS RATE
```

```
CC     UNITV(3,3):         MISSILE VELOCITY DIRECTION
CC     V(2,4):             MISSILE VELOCITY MAGNITUDE
CC   MODULES CALLED:  NONE.
CC   CALLING MODULES:  PRONAV(U)
CC
CC
CC
CC
      COMMON/FLAG/JPOL,KPOL,LPOL,MPOL,NPOL,ISTART,ILAUN,JATMOS,
     1INIT8(2),MANUVR,IPOST

      COMMON/VECTOR/R(3,6),RREL(3,6),V(3,6),VREL(3,6)
      COMMON/RATE/OMEGAR(3,4),OMEGAB(3,4),RDOT(3),VDOT(3),
     1 THDOT(3),PHIDOT(3)
      COMMON/UNIT/UNITV(3,3),UNITPV(3,3),UNITR(3,3),UNITL(
     13,3),UNITT(3,3),A(3,3),D(3,3),A1(3,3),D1(3,3),UNITP(
     23,3),UNITY(3,3)
      COMMON/INTEG/HDP,T,VAR(6),DER(6),TEMPS(2,6),VARSTR(6),
     1 STIME,TIME,HM,DT,DTMIN,NINT
      COMMON/CONST/G,RAD,TAU(3),DAT(100),IZERO
      COMMON/DELAY/ACOMAD(3),ACOMDD(3),DELON,DELAY1,DELAY2,
     1DELAY3,TCOMP,APDLAY,DFLAG,LAGOUT,SAVEA(2,5),SLAGON

      DIMENSION DUMV(3,3),OLOWR(3),SAVEWR(3),OLDDV(3),SAVEDV(3),
     1 OLDUV(3),SAVEUV(3),OLDV4(1),SAVEV4(1)

CC
CC   INITIALIZATION CODE:  THE FIRST TIME THROUGH BUSDLY,  CCC
CC     SET LAST STORED DATA EQUAL TO CURRENT DATA.  AFTER  CCC
CC     THE FIRST TIME, LAST STORED DATA AND CURRENT DATA   CCC
CC     WILL BE DIFFERENT..                                 CCC

       IF (ISTART .NE. 0) GO TO 50
       DO 10 I=1,3
       OLDWR(I)=OMEGAR(3,I)
       OLDDV(I)=DUMV(1,I)
```

```
      OLDDV(I)=DUMV(1,I)
      OLDDUV(I)=UNITV(2,I)
   10 CONTINUE
      OLCV4(1)=V(2,4)
   50 CONTINUE
CC
CC   EACH TIME THROUGH BUSDLY, TEMPORARYILY SAVE THE        CCC
CC   CURRENT VALUES OF ALL THE DATA.                        CCC
CC
      DO 60 I=1,3
      SAVEWR(I)=OMEGAR(3,I)
      SAVEDV(I)=DUMV(1,I)
      SAVEUV(I)=UNITV(2,I)
   60 CONTINUE
      SAVEV4(1)=V(2,4)
CC
CC   IF CURRENT VALUE OF LOS RATE ARRIVED BEFORE COMPUTA-   CCC
CC   TION TIME, USE IT.  IF NOT, USE THE LAST STORED VALUE  CCC
CC
       IF (DELAY1 .LE. TCOMP) GO TO 100
       DO  70 I=1,3
       OMEGAR(3,I)=OLDWR(I)
       DUMV(1,I)=OLDDV(I)
    70 CONTINUE
CC
CC   IF THE MISSILE VELOCITY DATA ARRIVED AT THE GUIDANCE   CCC
CC   COMPUTER BY COMPUTATION TIME, USE IT.  IF NOT, USE     CCC
CC   THE LAST STORED VALUES.                                CCC
CC
  100 IF (DELAY3 .LE. TCOMP) GO TO 150
```

```
      DO 120 I=1,3
      UNITV(2,I)=OLDUV(I)
  120 CONTINUE
      V(2,4)=OLDV(1)

CC
CC    STORE THE CURRENT DATA VALUES FOR USE AT THE NEXT DT  CCC
CC    TIME.                                                 CCC
CC

  150 DO 170 I=1,3
      OLDWR(I)=SAVEWR(I)
      OLDDV(I)=SAVEDV(I)
      OLDUV(I)=SAVEUV(I)
  170 CONTINUE
      OLDV4(1)=SAVEV4(1)
      RETURN
      END
```

```
****************************************
     SUBROUTINE DIGITAL
****************************************
CC   MODULE NAME:  DIGITAL
CC   FUNCTION:  THIS ROUTINE SETS CERTAIN VARIABLES USED IN  REVISED
CC     TACTICSIV.  THIS PROCESSINCLUDES FIRST INSURING THAT CERTAIN
CC     VARIABLES ARE SET TO MANDATORY TACTICSIV VALUES.  ADDITIONALLY,
CC     THIS ROUTINE INSURES THAT REVISED TACTICSIV DELAYS ARE WITHIN
CC     SPECIFIED REGIONS OF DEFINITION.  FINALLY, DIGITAL COMPUTES
CC     "APDLAY" WHICH IS USED IN APILOT.
CC   INPUTS:
CC     DATA(I):                  INPUT DATA ARRAY
CC   OUTPUTS:
CC     DATA(24):                 TAU(2) SET TO 0
CC     DATA(26):                 TAU1 SET TO .00001
CC     DATA(27):                 TAU2 SET TO .00001
CC     DATA(30):                 NINT SET TO 100
CC     DATA(38):                 DTMIN SET TO .0025
CC     DATA(43):                 DT SET TO .01
CC     DATA(45):                 IAUTO SET TO 1
CC     DELON                     DATA(80)
CC     DELAY1                    DATA(81)
CC     DELAY2                    DATA(82)
CC     DELAY3                    DATA(83)
CC     TCOMP                     DATA(84)
CC     SLAGON                    DATA(85)
CC     APDLAY                    APDLAY=TCOMP+DELAY2
CC     LAGOUT                    INITIALIZED TO ZERO
CC   MODULES CALLED:  NONE
CC   CALLING MODULES:  INCOND
CC
CC
CC
CC
     COMMON/CONST/G,RAD,TAU(3),DATA(100),IZERO
     COMMON/DELAY/ACOMAD(3),ACOMDD(3),DELON,DELAY1,DELAY2,
```

```
     1DELAY3,TCOMP,APDLAY,DFLAG,LAGOUT,SAVEA(2,5),SLAGON

      COMMON/INTEG/HDP,T,VAR(6),DER(6),TEMPS(2,6),VARSTR(6),
     1STIME,TIME,HM,DT,DTMIN,NINT
      COMMON/FLAG/JPOL,KPOL,LPOL,,MPOL,NPOL,ISTART,ILAUN,JATMOS,
     1INIT8(2),MANUVR,IPOST

CC
CC    INSURE THAT KEY DATA ITEMS ARE SET SET TO MANDATORY VALUES  CCC
CC

      IF (DATA(24) .EQ. 0) GO TO 5
      DATA(24)=0
      PRINT 200
    5 CONTINUE
      IF (DATA(26) .EQ. 0) GO TO 10
      DATA(26)=.00001
      PRINT 205
   10 CONTINUE
      IF (DATA(27) .EQ. 0) GO TO 15
      DATA(27)=.00001
      PRINT 205
   15 CONTINUE
      IF (DATA(30) .EQ. 100) GO TO 20
      DATA(30)=100
      PRINT 215
   20 CONTINUE
      IF(DATA(38).LT..0005)DATA(38)-.0005
      IF(DATA(38).GT..01)DATA(38)=.01
      IF (DATA(43) .EQ. .01) GO TO 30
      DATA(43)=.01
      PRINT 225
   30 CONTINUE
      IF (DATA(45) .EQ. 1) GO TO 35
      DATA(45)=1
      PRINT 230
```

```
   35 CONTINUE
CC
CC   DEFINE KEY VARIABLES CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC
      DELON=DATA(80)
      DELAY1=DATA(81)
      DELAY2=DATA(82)
      DELAY3=DATA(83)
      TCOMP=DATA(84)
      SLAGON=DATA(85)
CC
CC   INSURE DELAYS FALL WITHIN REGION OF DEFINITION  CCCC
CC
      IF (DELAY1 .GE. 0) GO TO 50
      DELAY1=0
   50 IF (DELAY1 .LE. .01) GO TO 55
      DELAY1=.01
   55 CONTINUE
      IF (DELAY2 .GE. 0) GO TO 60
      DELAY2=0
   60 IF (DELAY2 .LE. .005) GO TO 65
      DELAY2=.005
   65 CONTINUE
      IF (DELAY3 .GE. 0) GO TO 70
      DELAY3=0
   70 IF(DELAY3 .LE. .01) GO TO 75
      DELAY3=.01
   75 CONTINUE
      IF (TCOMP .GE. 0) GO TO 80
      TCOMP=0
   80 IF (TCOMP .LE. .005) GO TO 85
      TCOMP=.005
```

```
   85 CONTINUE

CC
CC   DEFINE APDLAY FOR USE BY APILOT(I) CCCCCCCCCCCCCCCCCC
CC

      APDLAY=TCOMP+DELAY2

CC
CC   INITIALIZE LAGOUT TO AID PRONAV IN THE LAST
CC   MOMENTS OF THE ENGAGEMENT.
      LAGOUT=0
CC

      DO 90 J=1,2
      DO 90 K=1,5
      SAVEA(J,K)=0
   90 CONTINUE
  100 CONTINUE
      RETURN

CC
CC   ERROR MESSAGE FORMATS CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC
 200 FORMAT( " WARNING:TAU(2) SET TO ZERO FOR REVISED TACTICSIV")
 205 FORMAT( " WARNING:APILOT LEAD/LAG SET TO ONE FOR  REVISED TACTICSIV
    1")
 215 FORMAT( " WARNING:NINT SET TO 100 FOR REVISED TACTICSIV")
 220 FORMAT( " WARNING:DTMIN SET TO .0025 FOR REVISED TACTICSIV")
 225 FORMAT( " WARNING:DT SET TO .01 FOR REVISED TACTICSIV")
 230 FORMAT( " WARNING:IAUTO SET TO 1 FOR REVISED TACTICSIV")
     END
```

```
CC****************************************
     SUBROUTINE   GCNOIS
CC****************************************
CC
CC   MODULE NAME:   GCNOIS
CC   FUNCTION:  THIS ROUTINE INJECTS A MOISE INTO THE COMMANDED
CC     ACCELERATION.  THIS NOISE RESULTS FROM  THE FINITE WORD
CC     LENGTH OF THE GUIDANCE COMPUTER.  THE NOISE IS ZERO MEAN
CC     NORMALLY DISTRIBUTED.  THE STANDARD DEVIATION IS DETERMINED
CC     BY THE NUMBER OF BITS OF THE COMPUTER.
CC   INPUTS:
CC     ACOM(2):                COMMANDED ACCELERATION
CC     ASMAX(2):               MAX G'S MISSILE CAN FULL
CC   OUTPUTS:
CC     ACOM(2):                COMMANDED ACCELERATION
CC   MODULES CALLED:
CC     ANRMRN:                 NORMAL DISTRIBUTION GENERATION FUNCTION
CC   CALLING MODULES:
CC     PRONAV
CC
CC
      REAL GCSIG,GCERR,K,M,N,Q
      COMMON/AERO/THRUST(3),ALPHA(3),CODRAG(3),CCLIFT(3),DRAG(3),
     1LIFT(3),WEIGHT(3),AREA(3),ASMAX(3),CLMAX(3),WO(3)
      COMMON/ACCEL/ACOM(3),ACOMA(3),ACOMD(3),AOUT(3),ACUTA(3),AOUTD(3)
     1,GFORCE(3),ACLMAX(3)
      M=ASMAX(2)
      N=16
      Q=M/(2**N)
      K=12
      GCSTG=Q/SQRT(K)
      GCERP=ANRMRN(GCSIG,0)
      IF(GCERR,GT,0)GO TO 50
      IF(ABS(SCERR).GT.(SQPT(K)*GCSIG))GCERR=(-1)*Q/2
      GO TO 100
   50 IF(ABS(GCERR).GT.(SQRT(K)*GCSIG))GCERR=Q/2
```

```
100 ACOM(2)=ACOM(2)*GCERR
    RETURN
    END
```

```
*****************************************
      SUBROUTINE LASTDT(LCASEX,LXDELT)
*****************************************
CC    MODULE NAME:  LASTDT(LCASEX,LXDELT)
CC    FUNCTION:  THIS ROUTINE ASSISTS APILOT IN COMPUTING THE
CC      AUTOPILOT RESPONSE DURING THE LAST OT TIME INTERVAL OF
CC      THE ENGAGEMENT.  AS MENTIONED IN APILOT(I), THE CURRENT
CC      OT TIME INTERVAL IS DIVIDED INTO TWO REGIONS.  THE FIRST
CC      EXTENDS FROM (TYME) TO (TYME+APDLAY).  THE SECOND REGION
CC      EXTENDS FROM (TYME+APOLAY) TO (TYME+DT).  IN THE FIRST
CC      REGION, ACOMD(I) DRIVES THE AUTOPILOT.  IN THE SECOND
CC      REGION, ACOM(I) DRIVES THE AUTOPILOT.  THE COMPLICATION IS
CC      THAT DURING THE LAST OT INTERVAL, "INTGRT" PRODUCES A
CC      TIME WINDOW (SMALLER THAN DT) WHICH IT USES TO SEARCH
CC      OUT THE PRECISE MOMENT THE ENGAGEMENT ENDED.  LASTDT FINDS
CC      WHAT PORTION OF THE WINDOW LIES IN THE ACOMD(I) REGION
CC      AND WHAT PORTION LIES IN THE ACOM(I) REGION.
CC    INPUTS:
CC      APDLAY:                 SIZE OF AUTOPILOT DELAY
CC      DT:                     BASIC INTEGRATION STEP SIZE
CC      HCP:                    CURRENT INTEGRATION  STEP SIZE
CC      NINT:                   NUMBER OF INTEGRATION STEPS
CC                              USED BY TRAPZ IN APILOT
CC      T:                      TIME AT THE BEGINNING OF HDP
CC    OUTPUTS:
CC      LCASEX:                 =1:HDP WINDOW AT LEAST PARTIALLY
CC                               WITHIN ACOMD(I) REGION
CC                              =2:HDP WINDOW COMPLETELY IN
CC                               ACOM(I) REGION
CC
CC      LXDELT:                 NUMBER OF TRAPZ STEPS WITHIN
CC                              ACOMD(I) REGION
CC
CC
      INTEGER LXDELT,LCASEX
      COMMON/ INTEG/ HDP,T,VAR(6),DER(6),TEMPS(2,6)VARSTR(6),
```

```
     1  STIME,TIME,HM,DT,DTMIN,NINT
      COMMON/DELAY/ACOMAD(3),ACOMDD(3),DELON,DELAY1,DELAY2,
     1DELAY3,TCOMP,APDLAY,DFLAG,LAGOUT,SAVEA(2,5),SLAGON

CC
CC    CHECK TO SEE IF HDP WINDOW IS COMPLETELY  IN CCCC
CC    THE ACOM REGION                              CCCC
CC

      TPOINT=T-(AINT(T/DT)=DT)
      IF (TPOINT .LT. APDLAY) GO TO 76
      LCASEX=2
      LXDELT=0
      GO TO 79

CC
CC    CHECK TO SEE IF HDP WINDOW IS COMPLETELY IN  CCCC
CC    THE ACOMD REGION                             CCCC
CC

   76 LCASEX=1
      IF ((TPOINT + HDP) .GT. APDLAY) GO TO 77
      LXDELT=NINT
      GO TO 79

CC
CC    CHECK TO SEE HOW MUCH OF THE WINDOW IS IN THE CCCC
CC    THE ACOMD REGION                              CCCC
CC

   77 DO 78 K=1,NINT
      TNINT=TPOINT + K=(HDP/NINT)
      IF (LXDELT .NE. 0) GO TO 78
      IF (TPOINT .LT. APDLAY) GO TO 78
      LXDELT=K
```

```
78 CONTINUE
79 CONTINUE
   RETURN
   END
```

```
CC***************************************
      SUBROUTINE SEKLAG
CC***************************************
CC
CC
CC    MODULE NAME:  SEKLAG
CC    FUNCTION:  THIS ROUTINE DELAYS LOS DATA .1 SECTONDS.
CC    INPUTS:
CC      ISTART:                  MAIN PROGRAM INITIALIZATION
CC                               FLAG
CC      OMEGAR:                  LOS RATE ARRAY
CC    OUTPUTS:
CC      OMEGAR:                  LOS RATE ARRAY
CC    MODULES CALLED:
CC      NONE.
CC    CALLING MODULES:
CC      PRONAV:
CC
CC
      INTEGER PTR
      COMMON /INTEG/ HDP,T,VAR(6),DER(6),TEMPS(2,6),VARSTR(6),
     1 STIME,TIME,HM,DT,DTMIN,NINT
      COMMON /RATE/ OMEGAR(3,4),OMEGAB(3,4),RDOT(3),VDOT(3),
     1 THDOT(3),PHIDOT(3)
      COMMON/FLAG/JPOL,KPOL,LPOL,MPOL,NPOL,ISTART,ILAUN,JATMOS,
     1INIT8(2),MANUVR,IPOST
      DIMENSION TEMPWR(4,10),STACKW(4)
      IF (ISTART .NE. 0) GO TO 10
CC
CC
CC    INITIALIZATION CODE
CC
      PTR=0
      DO 5 I=1,4
      STACKW(I)=0
```

```
    5 CONTINUE
      DO 7 I=1,4
      DO 7 J-1,10
      TEMPWR(T,J)=OMEGAR(3,*)
    7 CONTINUE
      GO TO 50
CC
CC    PTR CYCLES THROUGH TEN ARRAY COLUMNS EVERY .1 SECONDS.
CC
   10 PTR=PTR+1
      IF (PTR .NE. 11) GO TO 15
      PTR=1
   15 CONTINUE
CC
CC    SAVE CURRENT LOS RATE AND RETRIEVE LOS RATE
CC    FROM .1 SECONDS AGO.
CC
   25 DO 30 I-1,4
      STACKW(I)=TEMPWR(I,PTR)
      TEMPWR(I,PTR)=OMEGAR(3,I)
      OMEGAR(3,I)=STACKW(I)
   30 CONTINUE
   50 CONTINUE
      RETURN
      END
```

Appendix D

Test Run Graphs

Tactics IV produces miss distance and time of flight statistics. These statistics were plotted to provide for ease of reading. These plots are referred to as test run graphs and are divided into fixed launch range and variable launch range graphs. The two types of graphs will now be presented.

The fixed launch range graphs (Figure E-1 through E-24) depict mean miss distance versus Apdlay. Additionally, mean time of flight is included beside the miss distance data point. The variables in these test runs include attack geometry, missile modeling (deterministic or stochastic), basic integration step size (DT) and target maneuvering. These variables along with the figure numbers are shown in Table E-I.

The variable launch range graphs (Figure E-25 through E-30) depict mean miss distance versus launch range. Additionally, the Apdlay information is indicated symbolically. The variables in the test runs include missile modeling (deterministic or stochastic) and target maneuvering. In this set of tests, only the tail attack geometry and a DT of .01 seconds were considered. The graphs in this group are arranged as in Table E-II.

Table D-I

Fixed Launch Range Tests

Figure Number	Attack Geometry	Missile Model	DT	Apdlay	Target Maneuver
D-1	Tail	Deterministic	.01 sec	Five test Points*	Str/Lvl
D-2	Tail	Deterministic	.01 sec	Five test Points*	Turning
D-3	Tail	Deterministic	.01 sec	Five test Points*	Climb/ Dive
D-4	Frontal	Deterministic	.01 sec	Five test Points*	Str/Lvl
D-5	Frontal	Deterministic	.01 sec	Five test Points*	Turning
D-6	Frontal	Deterministic	.01 sec	Five test Points*	Climb/ Dive
D-7	Initial Heading Error	Deterministic	.01 sec	Five test Points*	Str/Lvl
D-8	Initial Heading Error	Deterministic	.01 sec	Five test Points*	Turning
D-9	Initial Heading Error	Deterministic	.01 sec	Five test Points*	Climb/ Dive
D-10	Tail	Deterministic	.1 sec	Three test Points**	Str/Lvl
D-11	Tail	Deterministic	.1 sec	Three test Points**	Turning
D-12	Tail	Deterministic	.1 sec	Three test Points**	Climb/ Dive
D-13	Tail	Stochastic	.01 sec	Five test Points**	Str/Lvl

*Five test points of Apdlay were 0, .0025, .005, .0075, .01 seconds.
**Three test points of Apdlay were 0, .005 and .01 seconds.
Note: Launch range was 10,000 feet in all these runs.

Table D-I (Cont'd)

Fixed Launch Range Tests

Figure Number	Attack Geometry	Missile Model	DT	Apdlay	Target Maneuver
D-14	Tail	Stochastic	.01 sec	Five test Points*	Turning
D-15	Tail	Stochastic	.01 sec	Five test Points*	Climb/ Dive
D-16	Frontal	Stochastic	.01 sec	Five test Points*	Str/Lvl
D-17	Frontal	Stochastic	.01 sec	Five test Points*	Turning
D-18	Frontal	Stochastic	.01 sec	Five test Points*	Climb/ Dive
D-19	Initial Heading Error	Stochastic	.01 sec	Five test Points*	Str/Lvl
D-20	Initial Heading Error	Stochastic	.01 sec	Five test Points*	Turning
D-21	Initial Heading Error	Stochastic	.01 sec	Five test Points*	Climb/ Dive
D-22	Tail	Stochastic	.1 sec	Three test Points**	Str/Lvl
D-23	Tail	Stochastic	.1 sec	Three test Points**	Turning
D-24	Tail	Stochastic	.1 sec	Three test Points**	Climb/ Dive

*Five test points of Apdlay were 0, .0025, .005, .0075, .01 seconds.
**Three test points of Apdlay were 0, .005 and .01 seconds.
Note: Launch range was 10,000 feet in all these runs.

Table D-II

Variable Launch Range Tests

Figure Number	Attack Geometry	Missile Model	DT	Apdlay	Target Maneuver
D-25	Tail	Deterministic	.01 sec	Three test Points**	Str/Lvl
D-26	Tail	Deterministic	.01 sec	Three test Points**	Turning
D-27	Tail	Deterministic	.01 sec	Three test Points**	Climb/ Dive
D-28	Tail	Stochastic	.01 sec	Three test Points**	Str/Lvl
D-29	Tail	Stochastic	.01 sec	Three test Points**	Turning
D-30	Tail	Stochastic	.01 sec	Three test Points**	Climb/ Dive

**Three test points of Apdlay were 0, .005 and .01 seconds.
Note: Launch range was set to 5Kft, 7.5Kft, 10Kft, 12.5Kft and 15Kft.

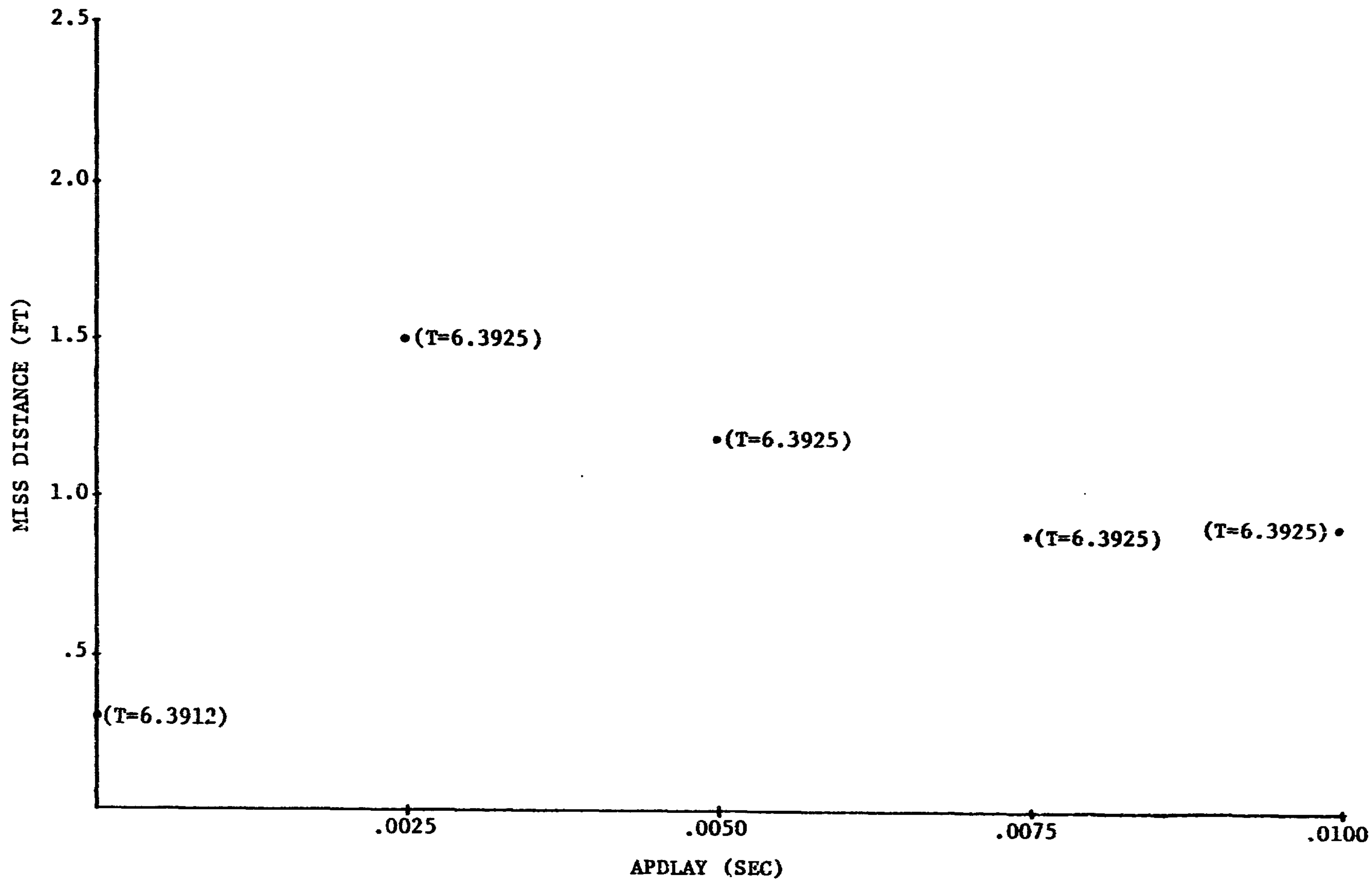


Figure D-1. Tail Attack, Str/Lvl Tgt, Deterministic, DT = .01 sec

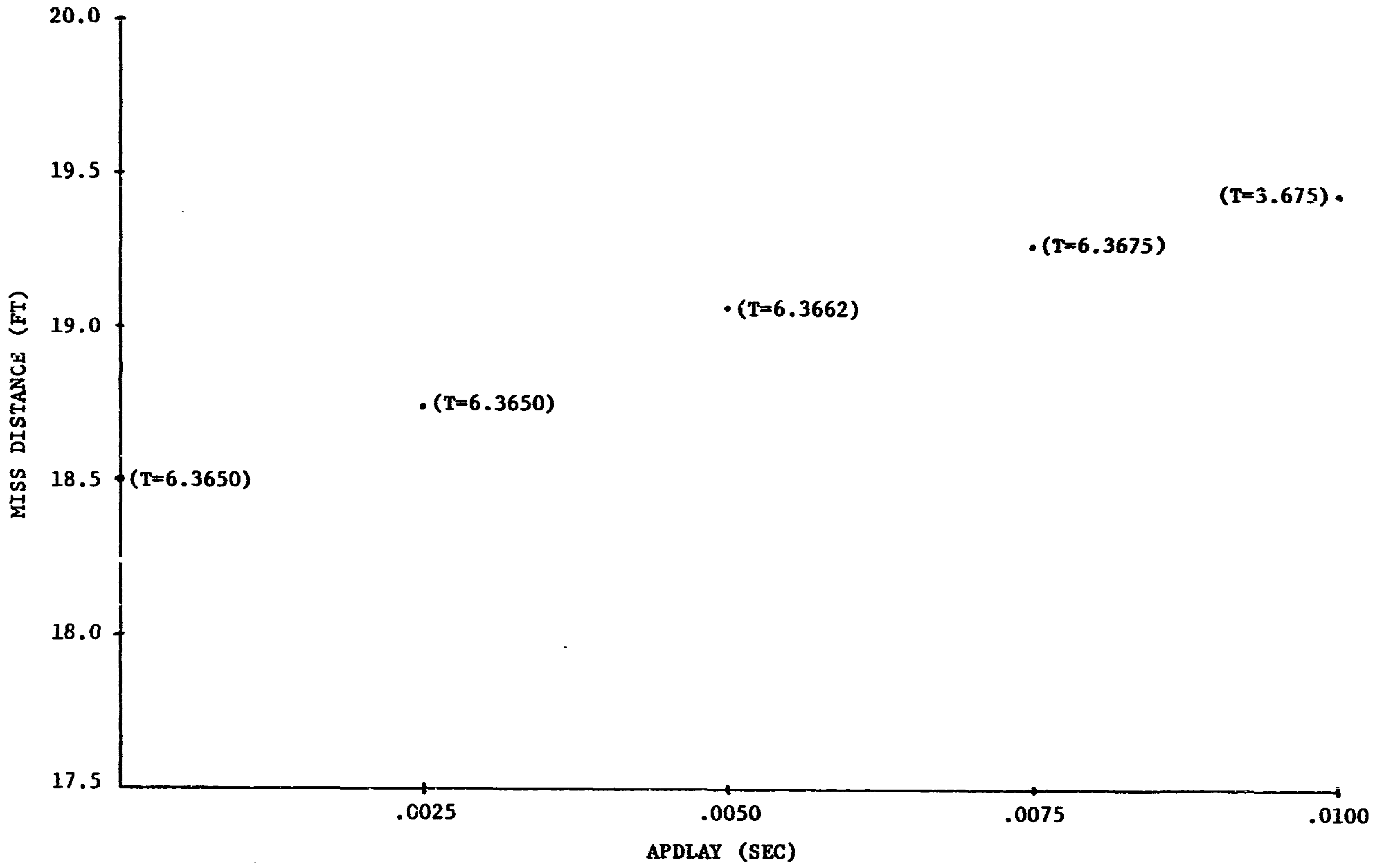


Figure D-2. Tail Attack, Turning Tgt, Deterministic, DT = .01 sec

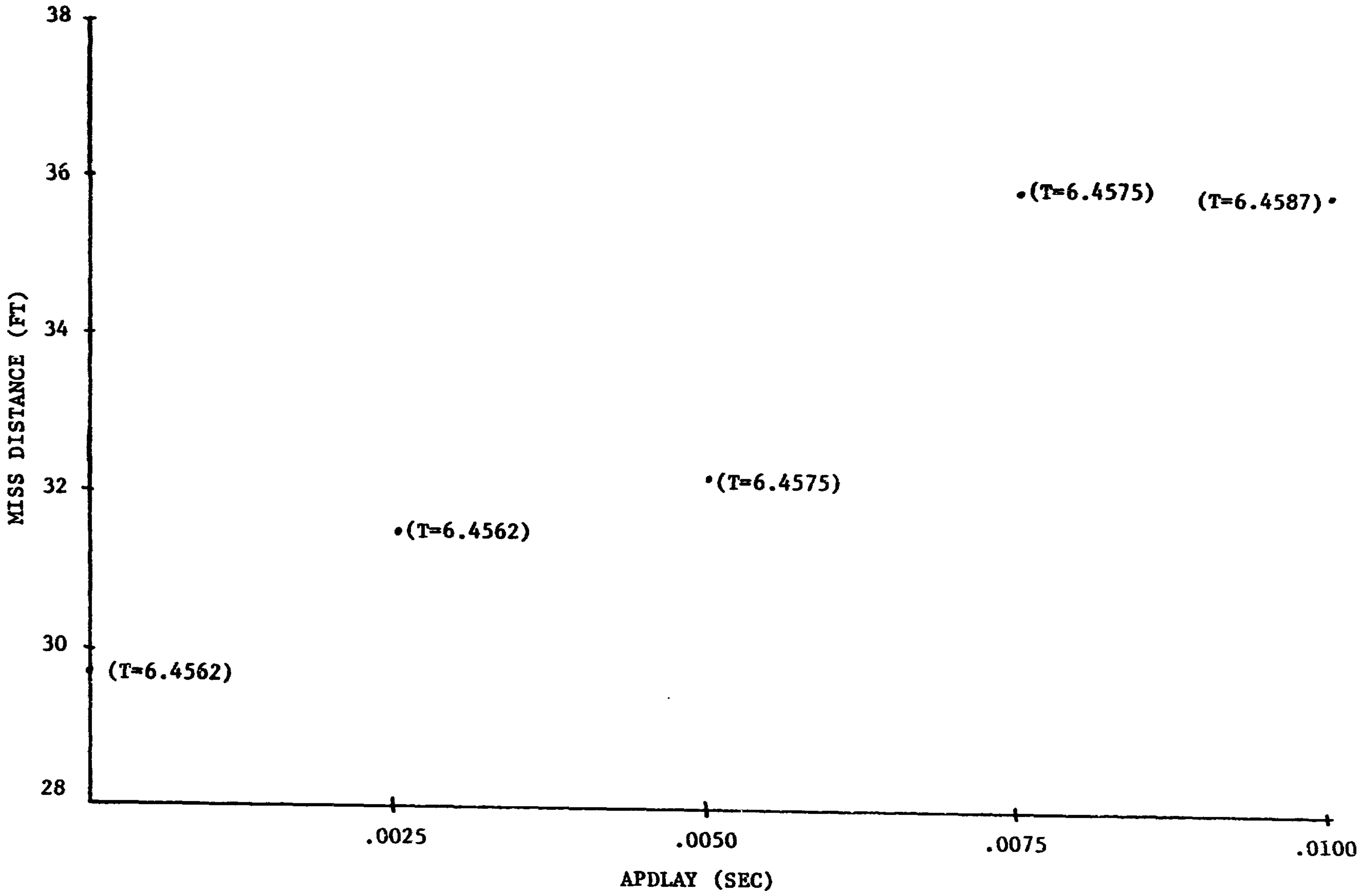


Figure D-3. Tail Attack, Climb/Dive Tgt, Deterministic, DT = .01 sec

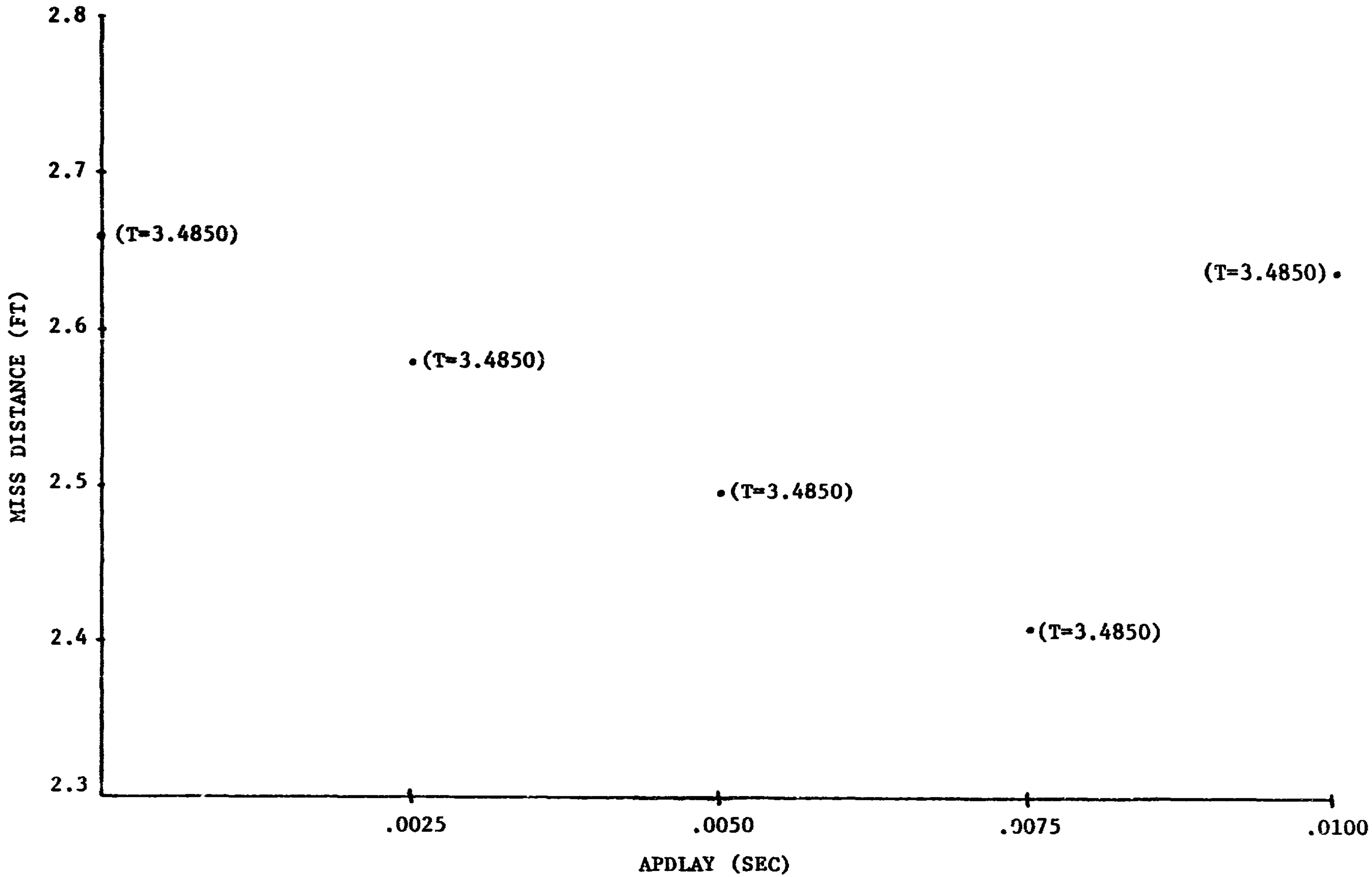


Figure D-4. Frontal Attack, Str/Lvl Tgt, Deterministic, DT = .01 sec

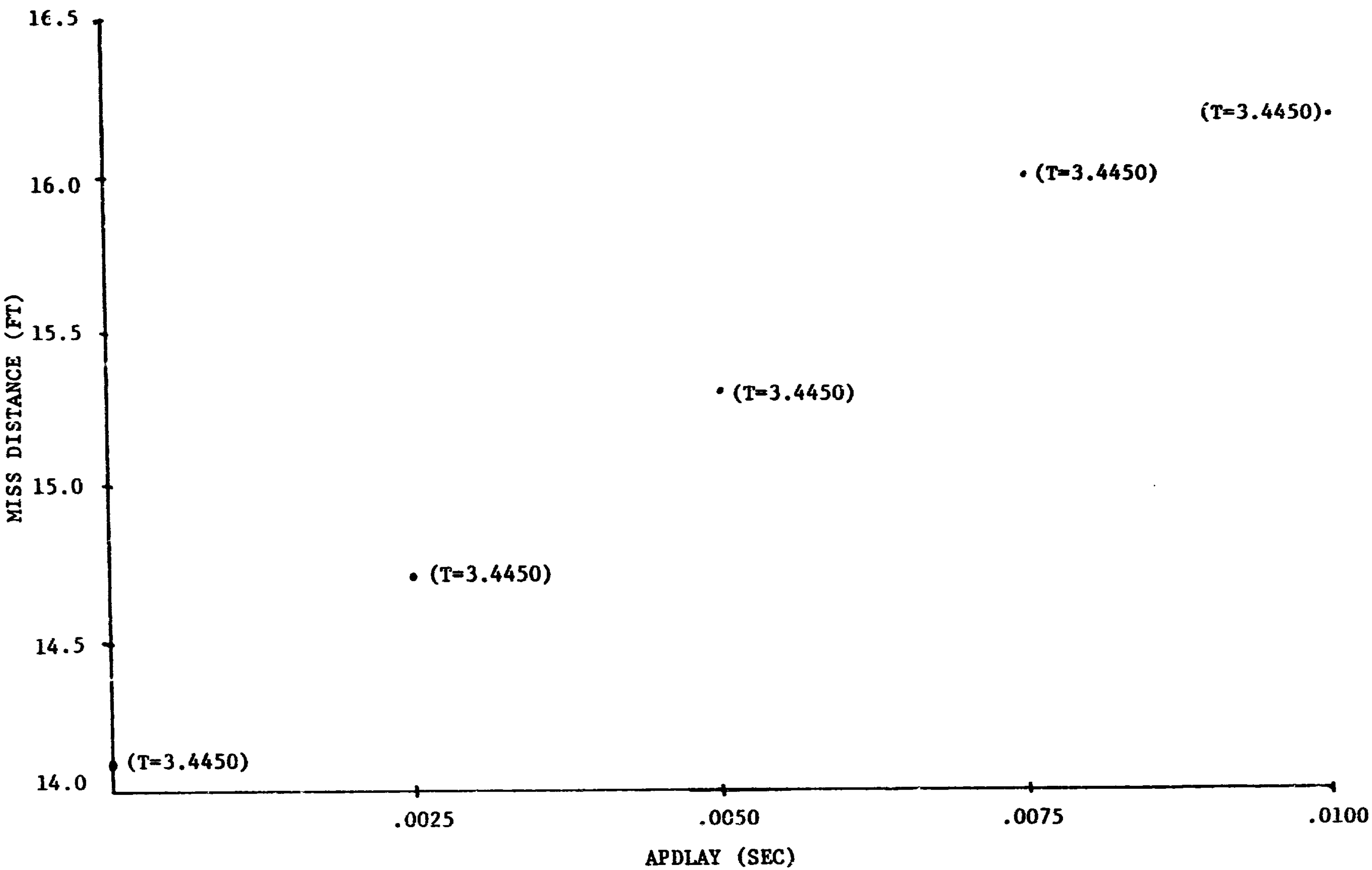


Figure D-5. Frontal Attack, Turning Tgt, Deterministic, DT = .01 sec

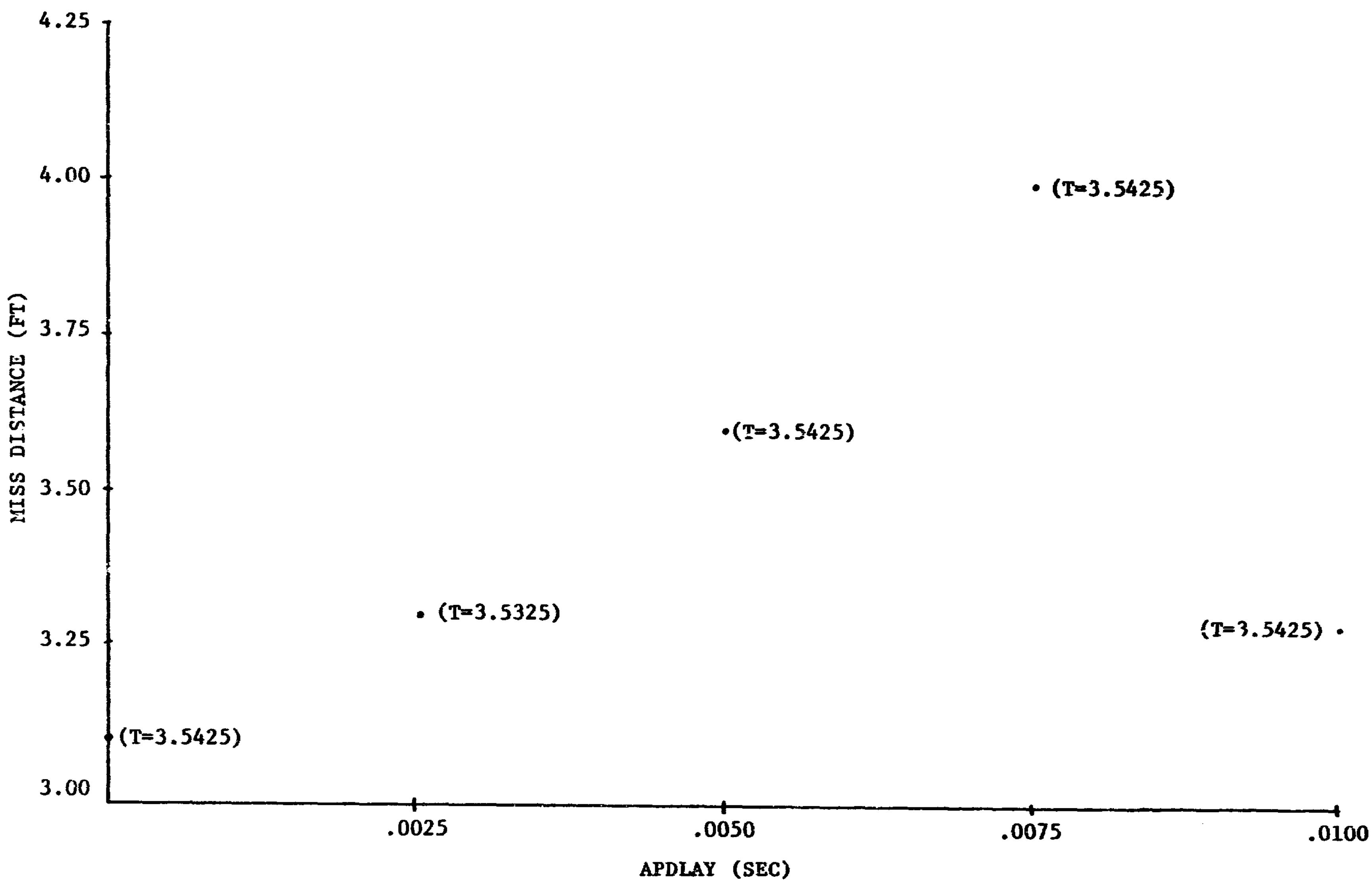


Figure D-6. Frontal Attack, Climb/Dive Tgt, Deterministic, DT = .01 sec

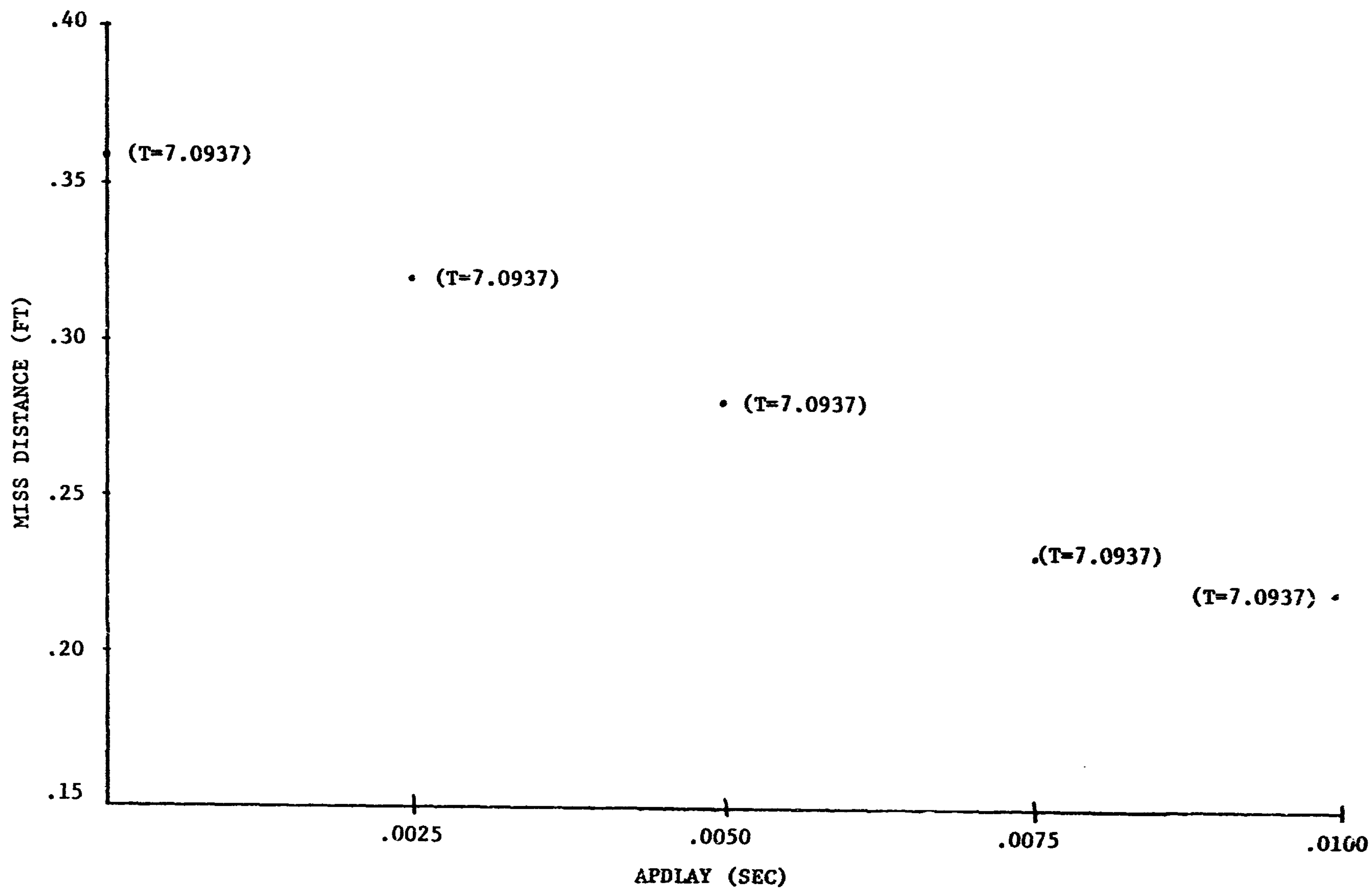


Figure D-7. Initial Heading Error, Tail Attack, Str/Level Tgt, Deterministic, DT = .01 sec

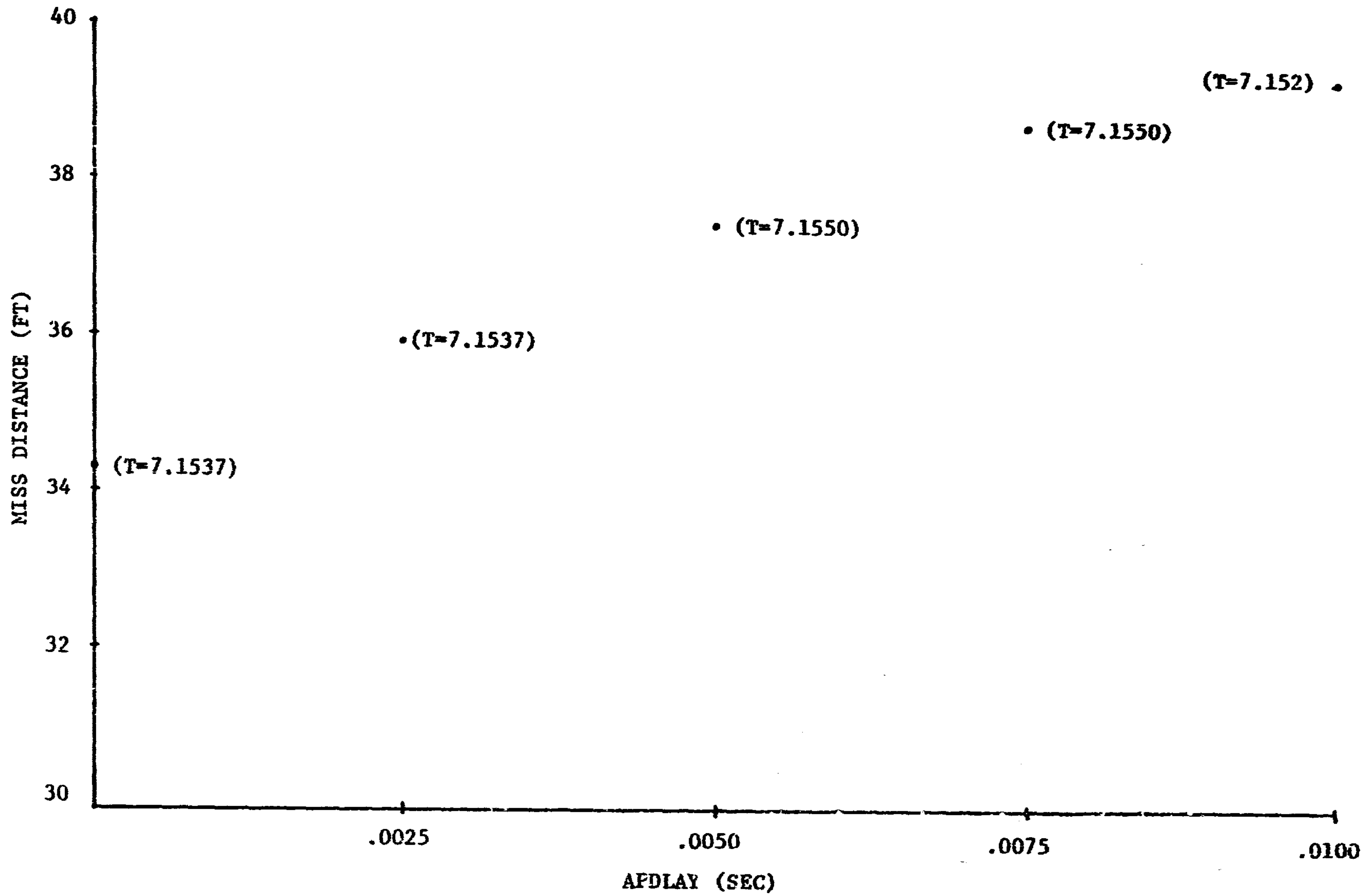


Figure D-8. Initial Heading Error, Tail Attack, Turning Tgt, Deterministic, DT = .01 sec

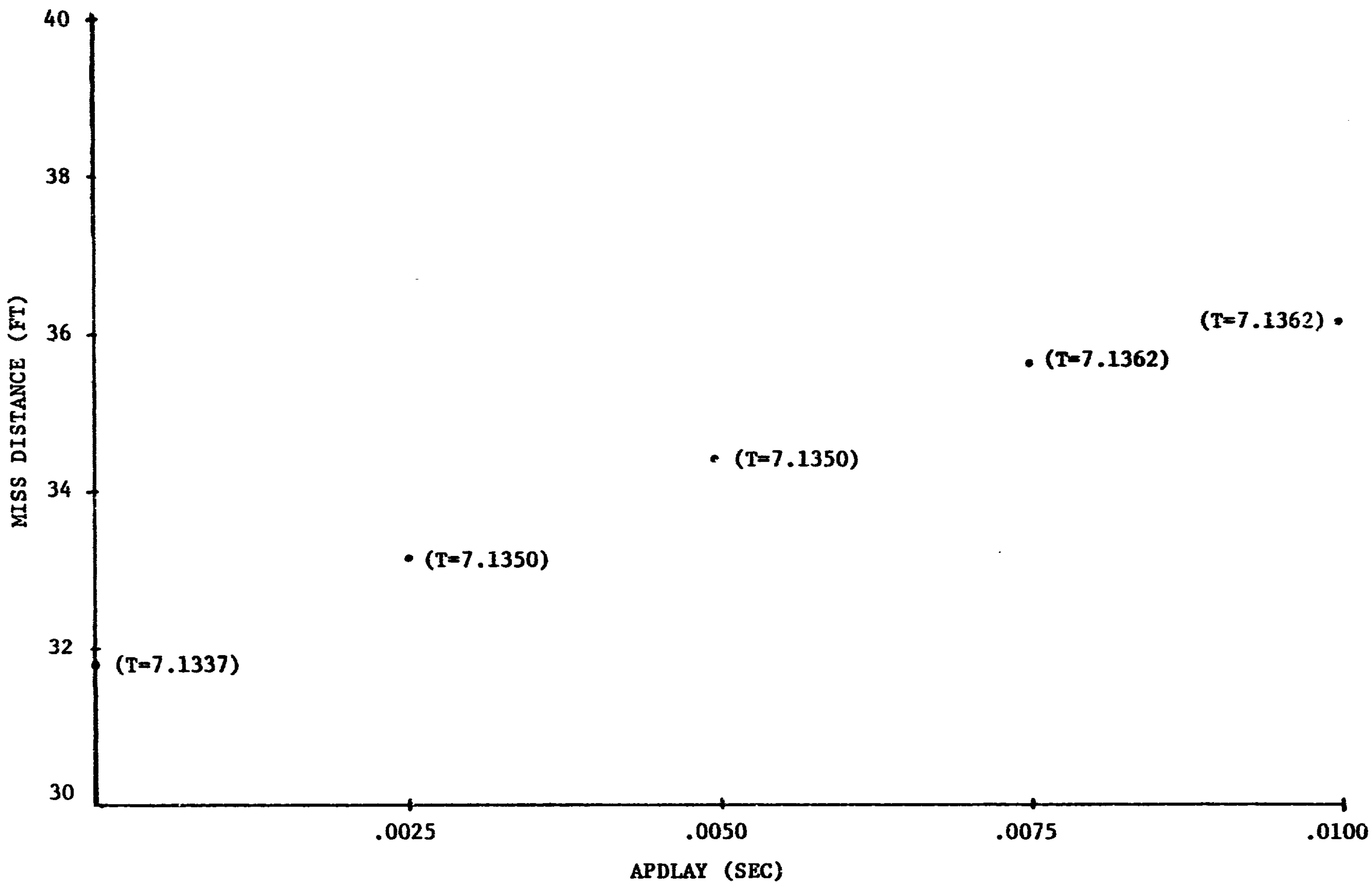


Figure D-9. Initial Heading Error, Tail Attack, Climb/Dive Tgt, Deterministic Deterministic, DT = .01 sec

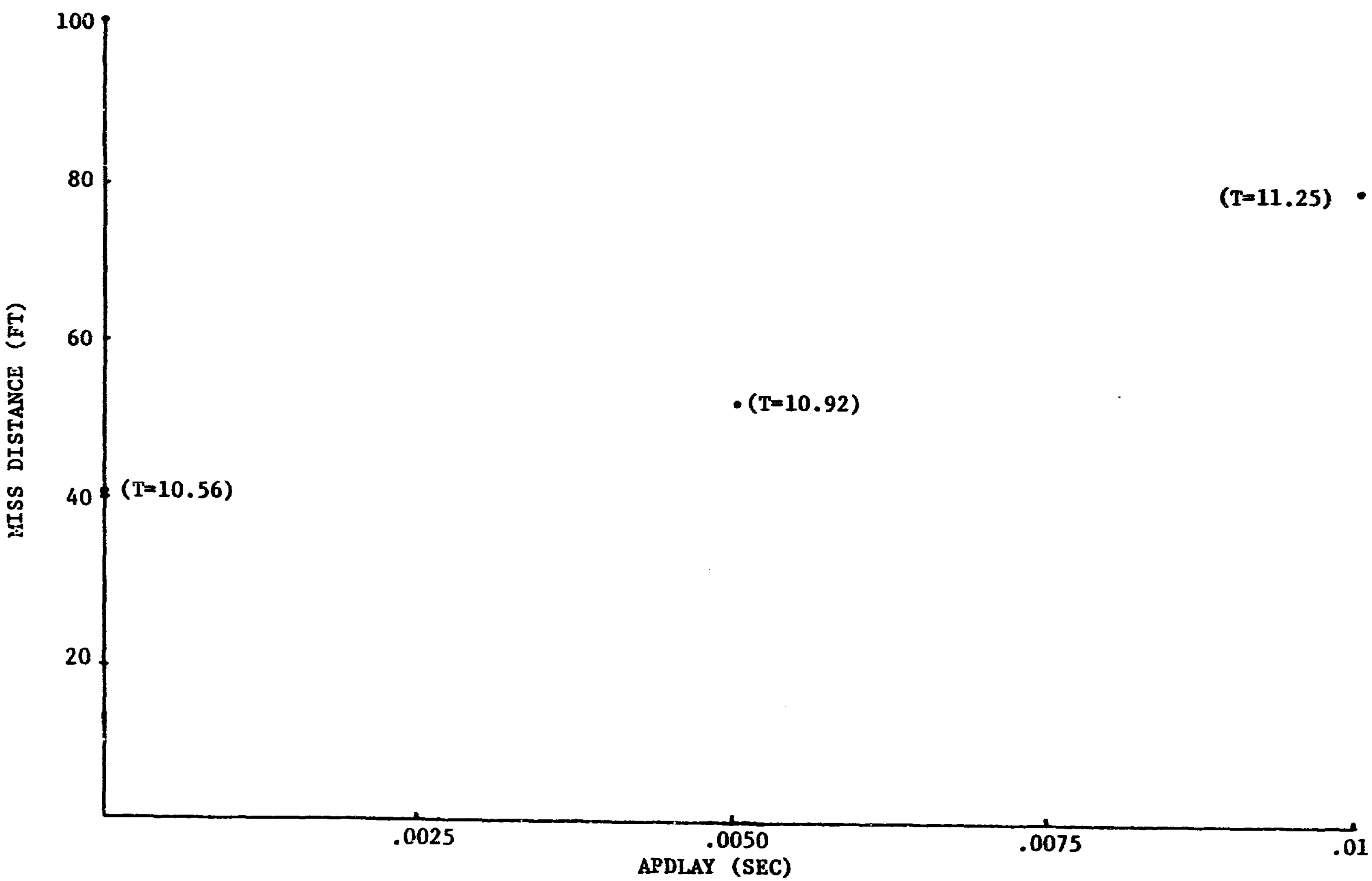


Figure D-10. Tail Attack, Str/Lvl Tgt, Deterministic, DT = .1 sec

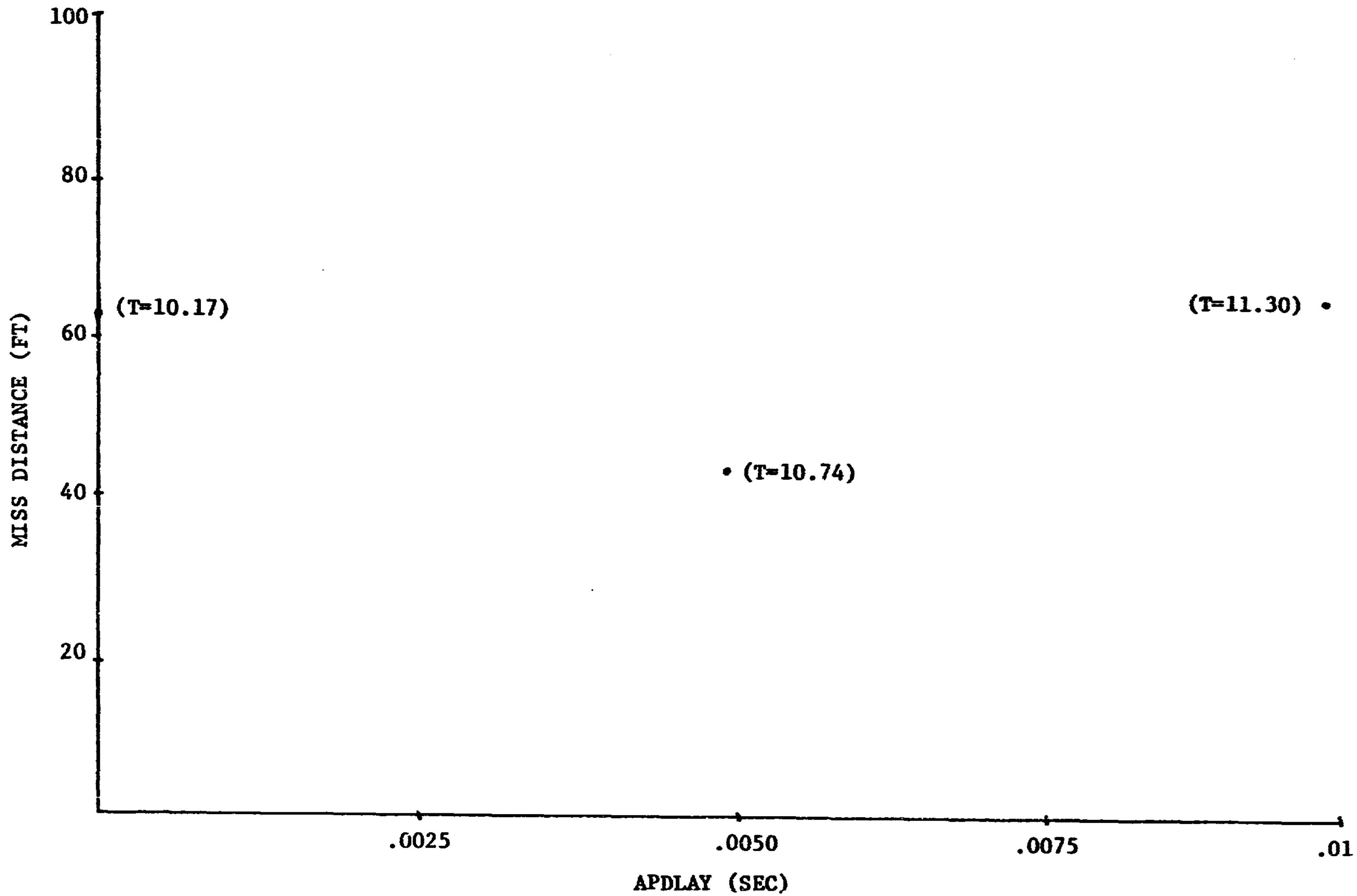


Figure D-11. Tail Attack, Turning Tgt, Deterministic, DT = .1 sec

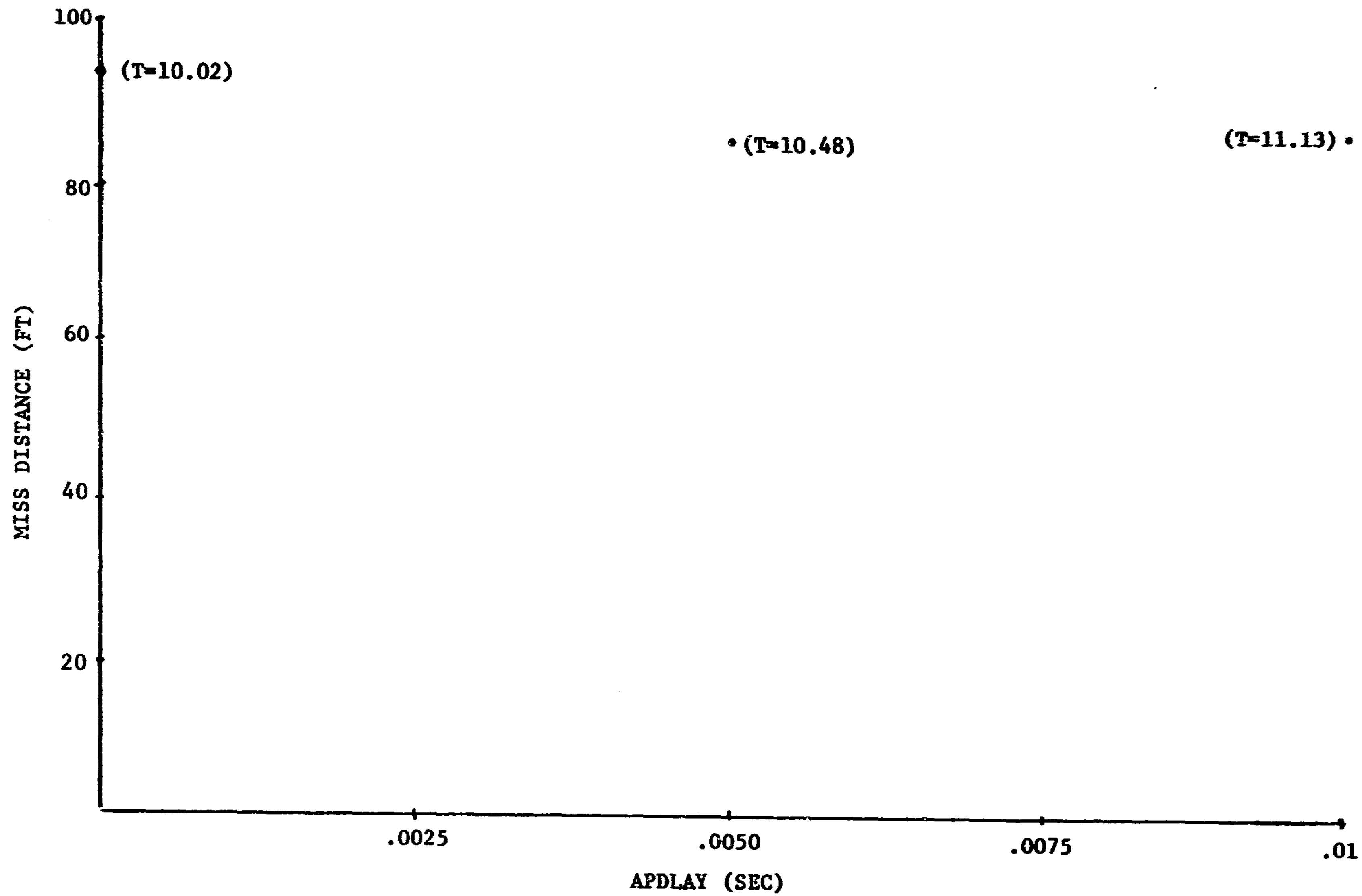


Figure D-12. Tail Attack, Climb/Dive Tgt, Deterministic, DT = .1 sec

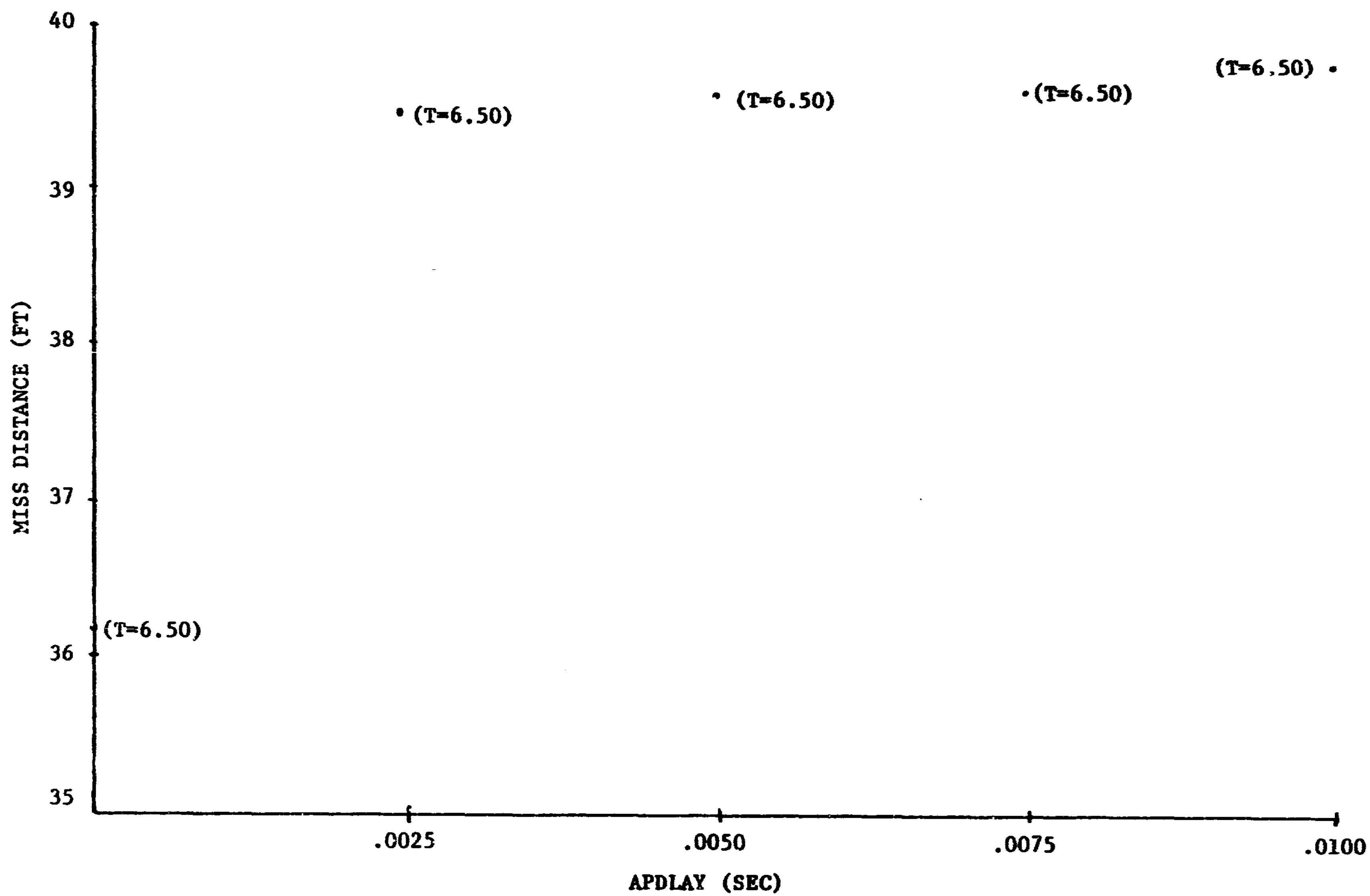


Figure D-13. Tail Attack, Str/Lvl Tgt, Stochastic, DT = .01 sec

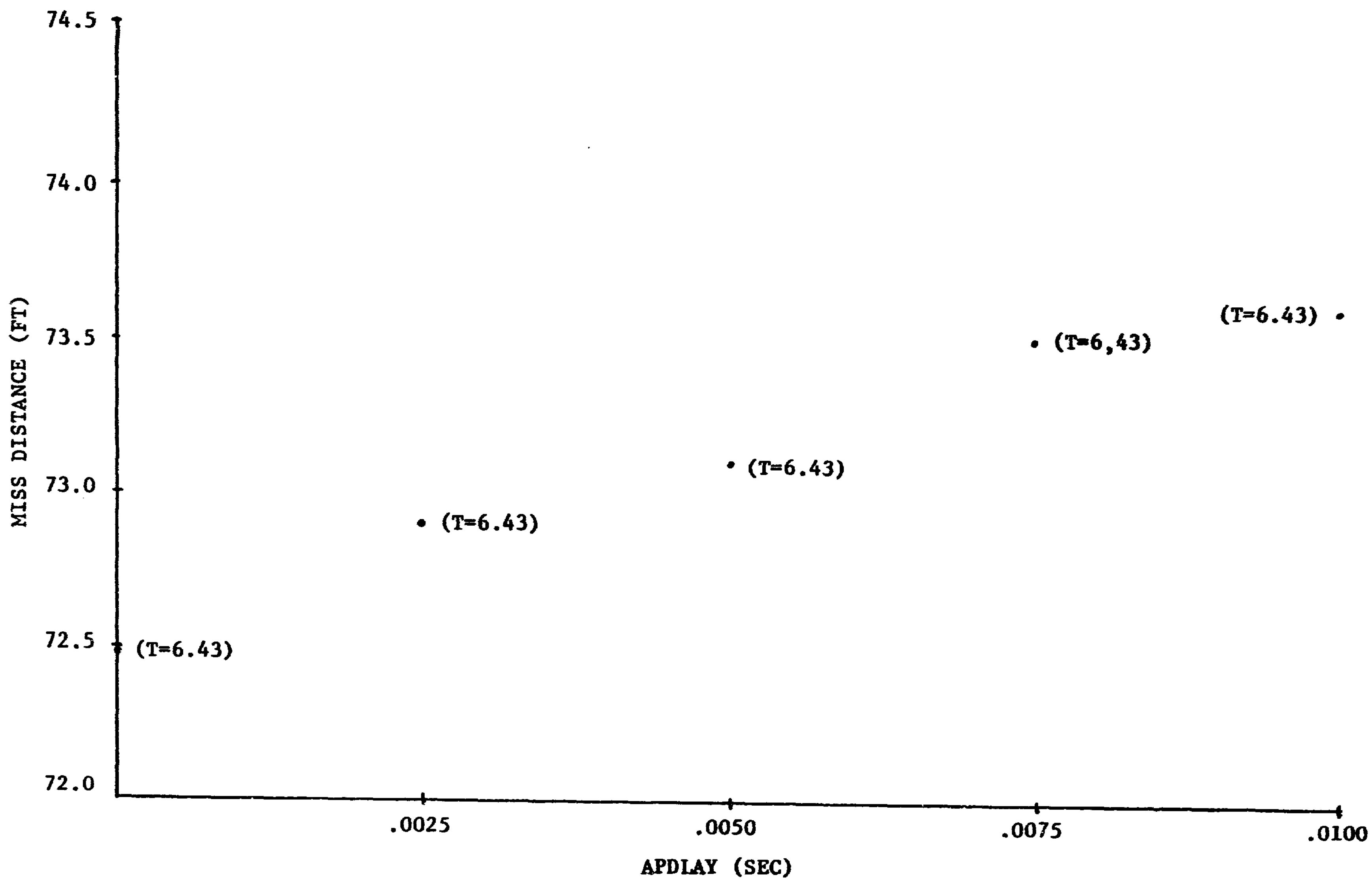


Figure D-14. Tail Attack, Turning Tgt, Stochastic, DT = .01 sec

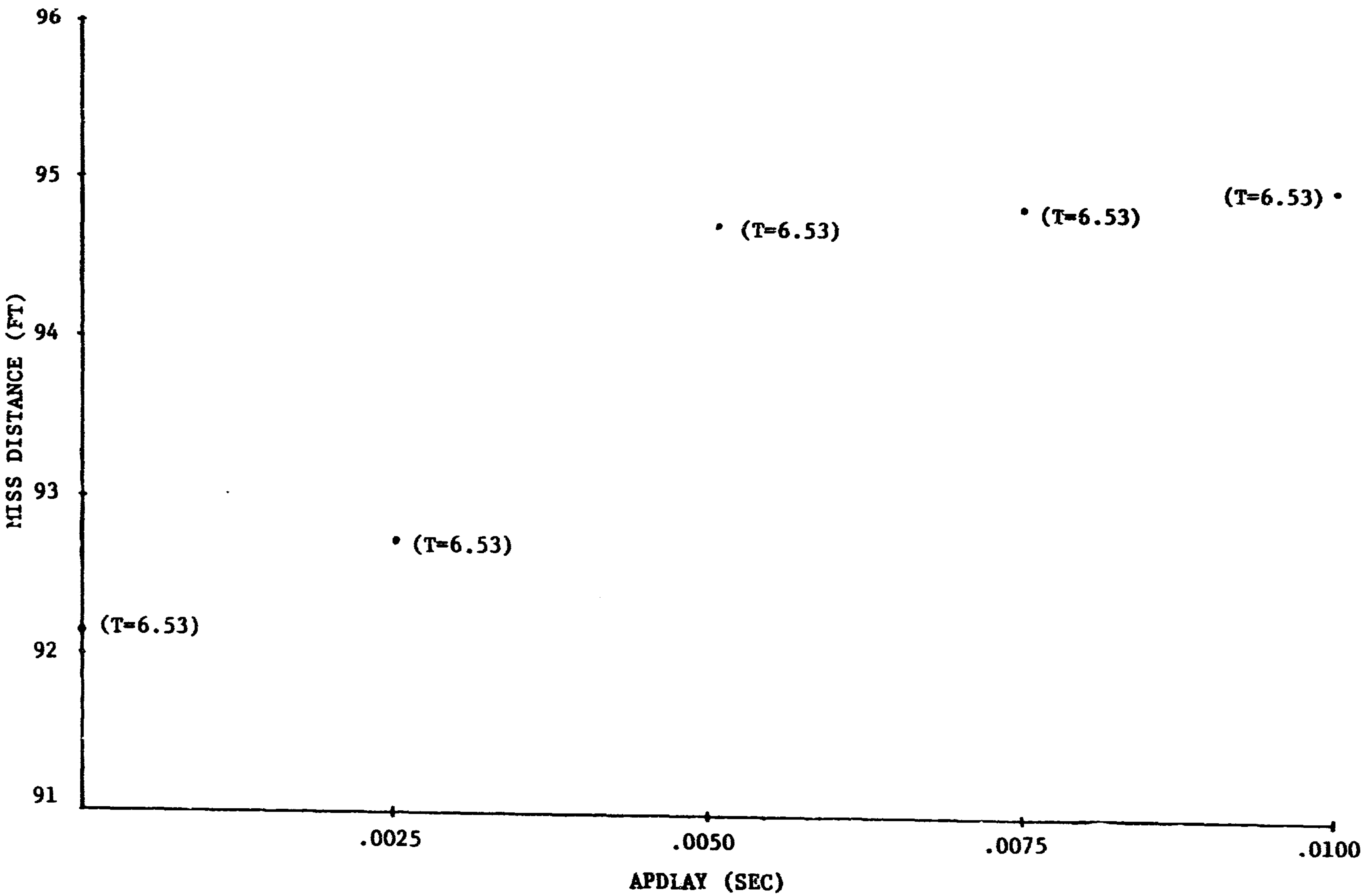


Figure D-15. Tail Attack, Climb/Dive Tgt, Stochastic, DT = .01 sec

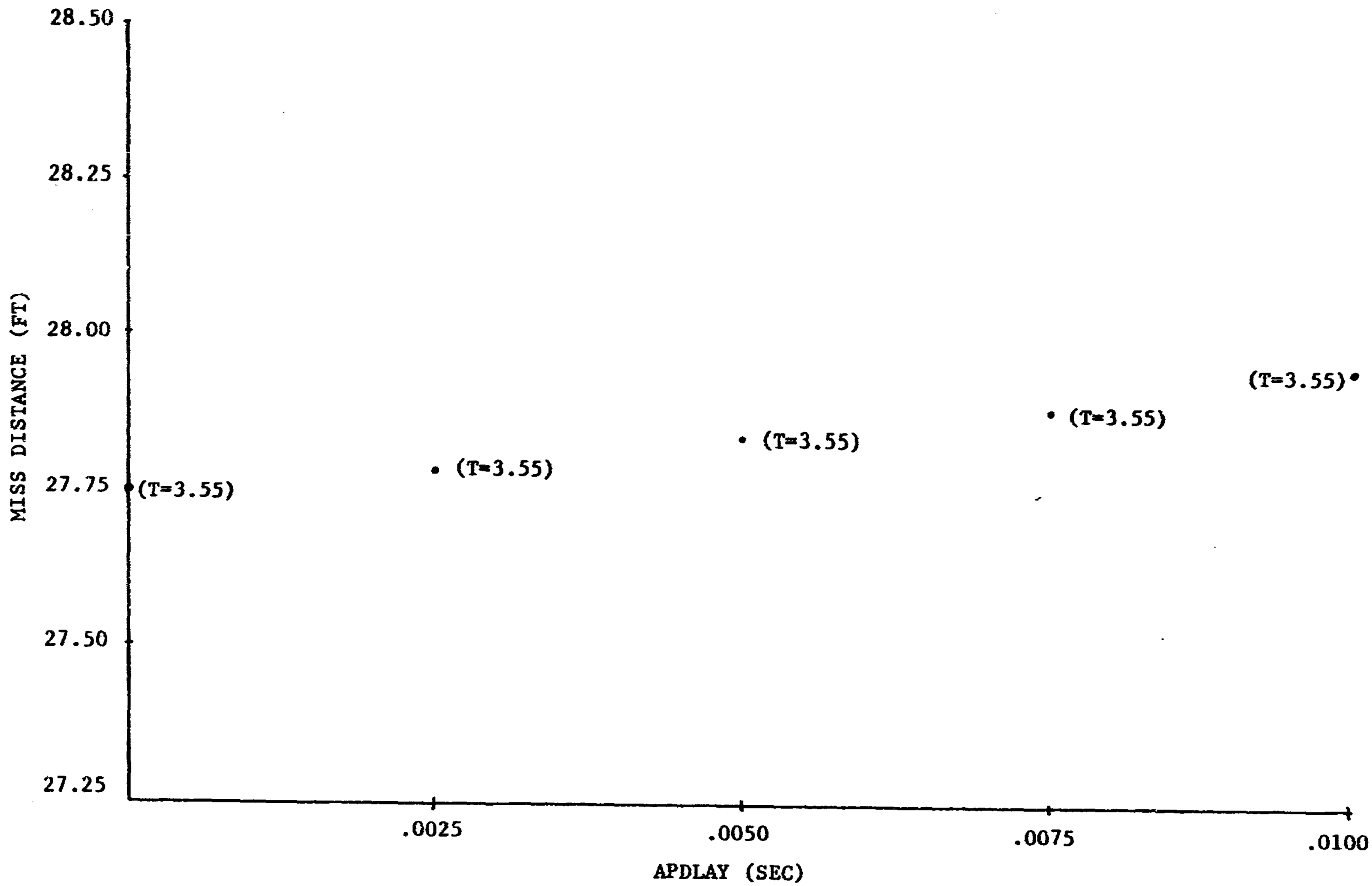


Figure D-16. Frontal Attack, Str/Lvl Tgt, Stochastic, DT = .01 sec

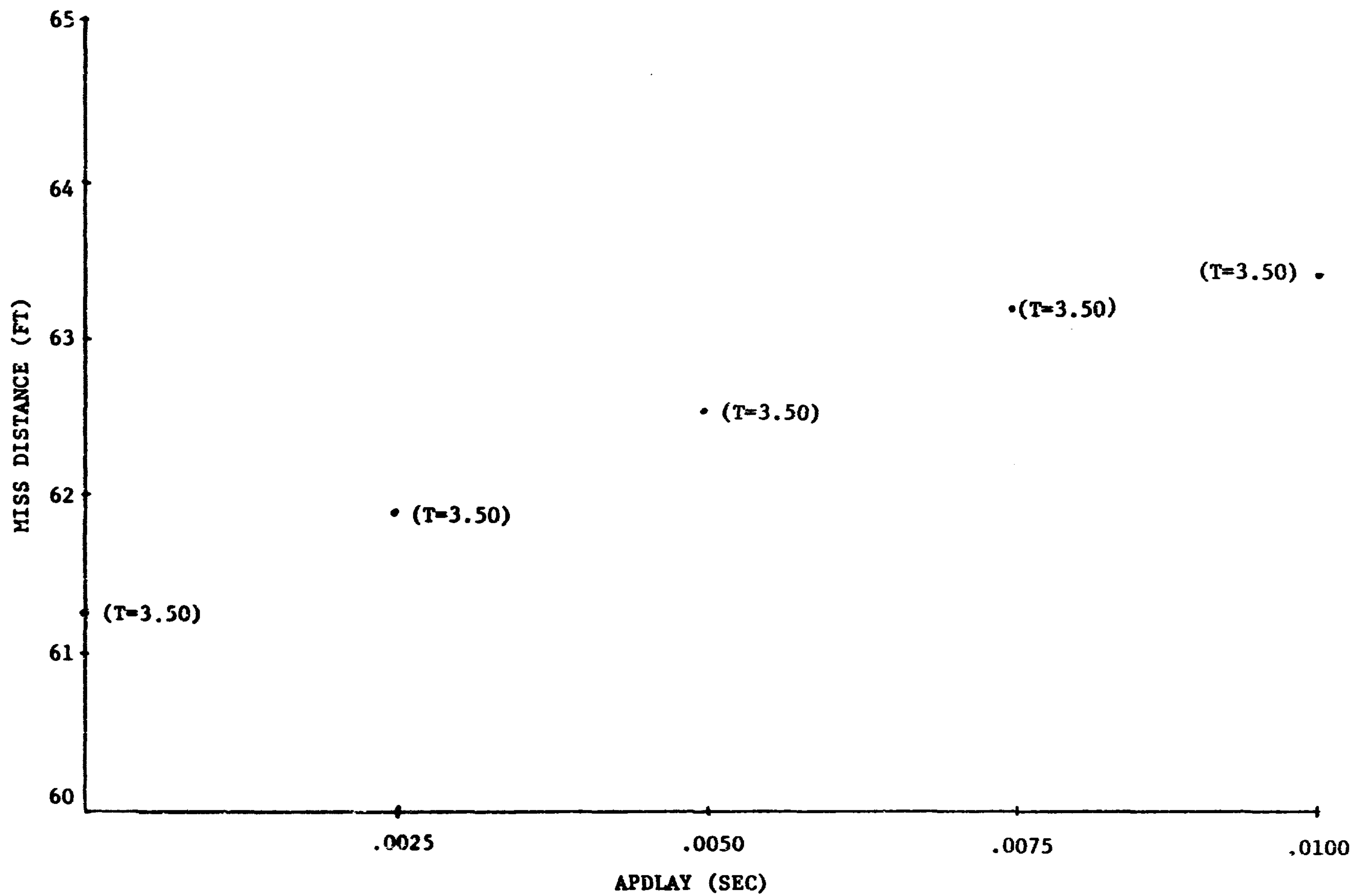


Figure D-17. Frontal Attack, Turning Tgt, Stochastic, DT = .01 sec

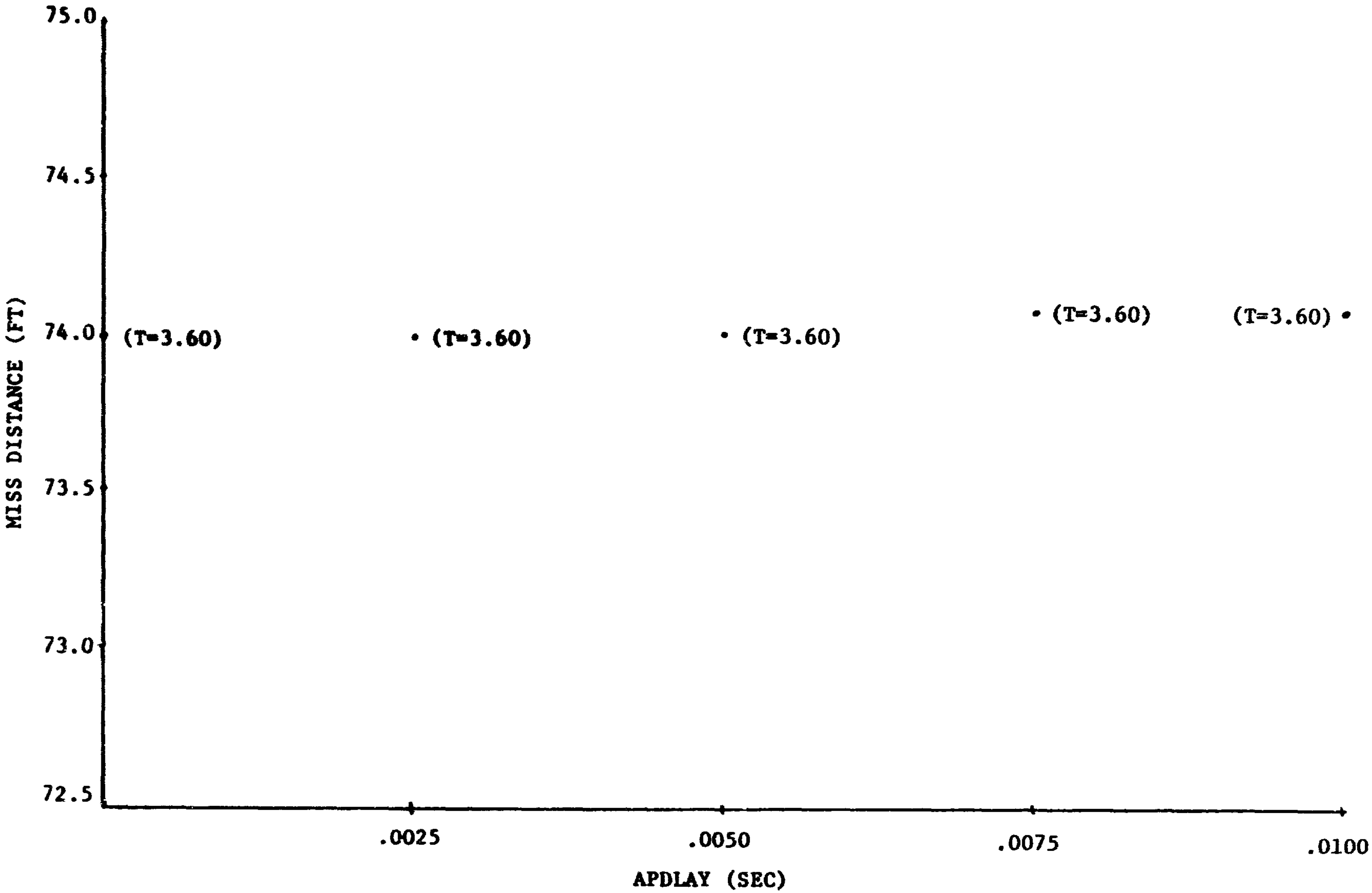


Figure D-18. Frontal Attack, Climb/Dive Tgt, Stochastic, DT = .01 sec

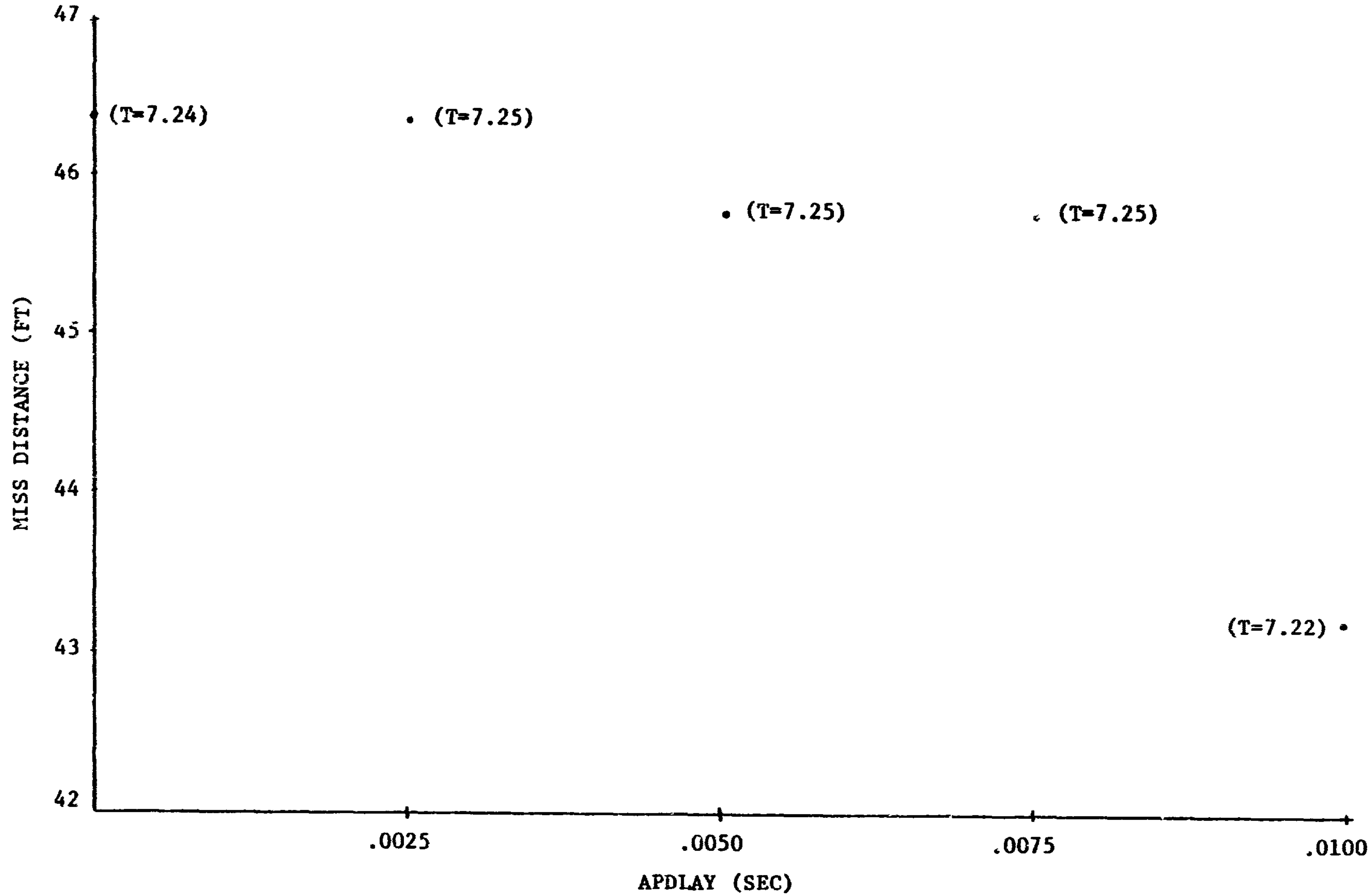


Figure D-19. Initial Heading Error, Str/Lvl Tgt, Stochastic, DT = .01 sec

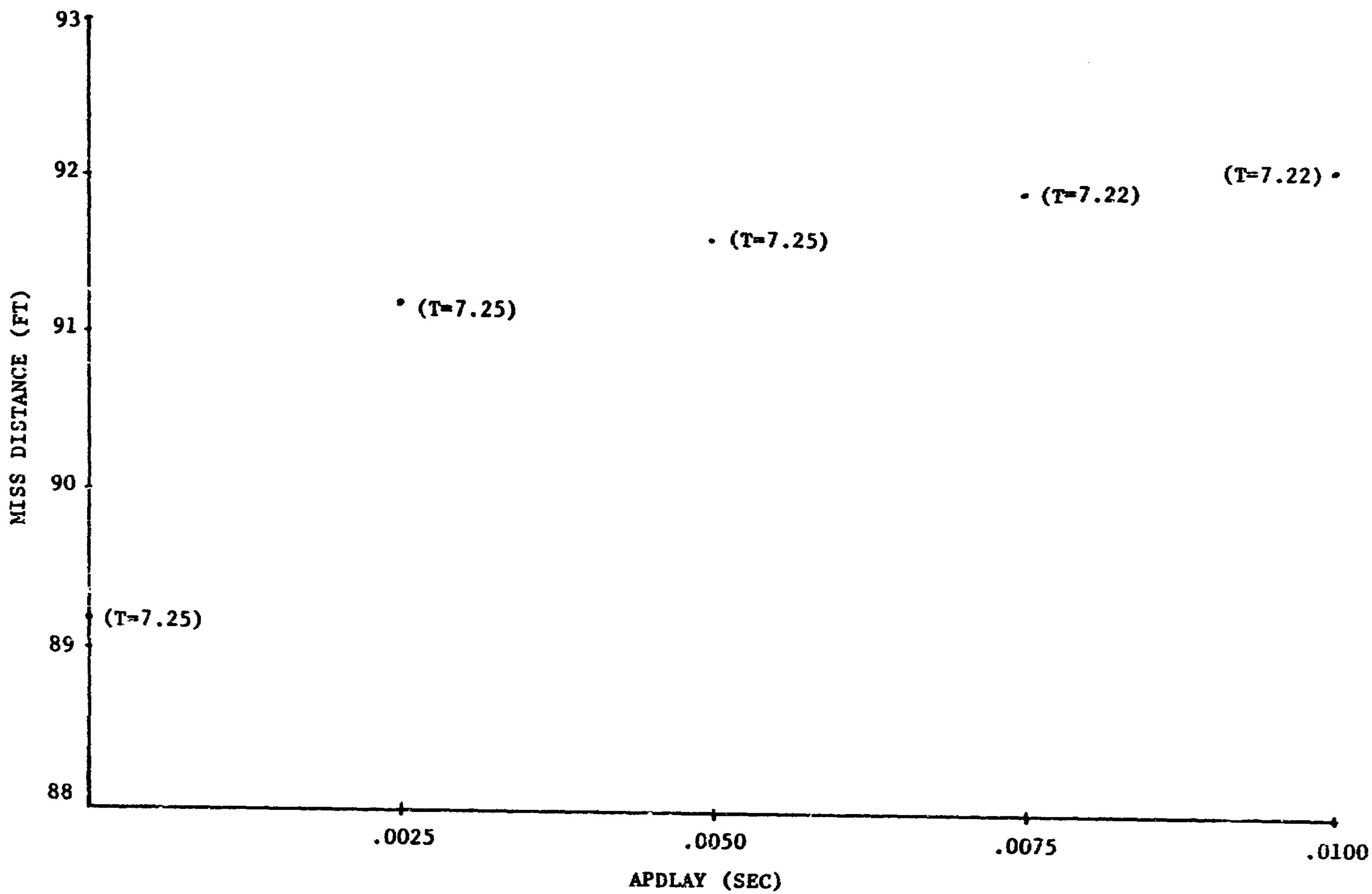


Figure D-20. Initial Heading Error, Turning Tgt, Stochastic, DT = .01 sec

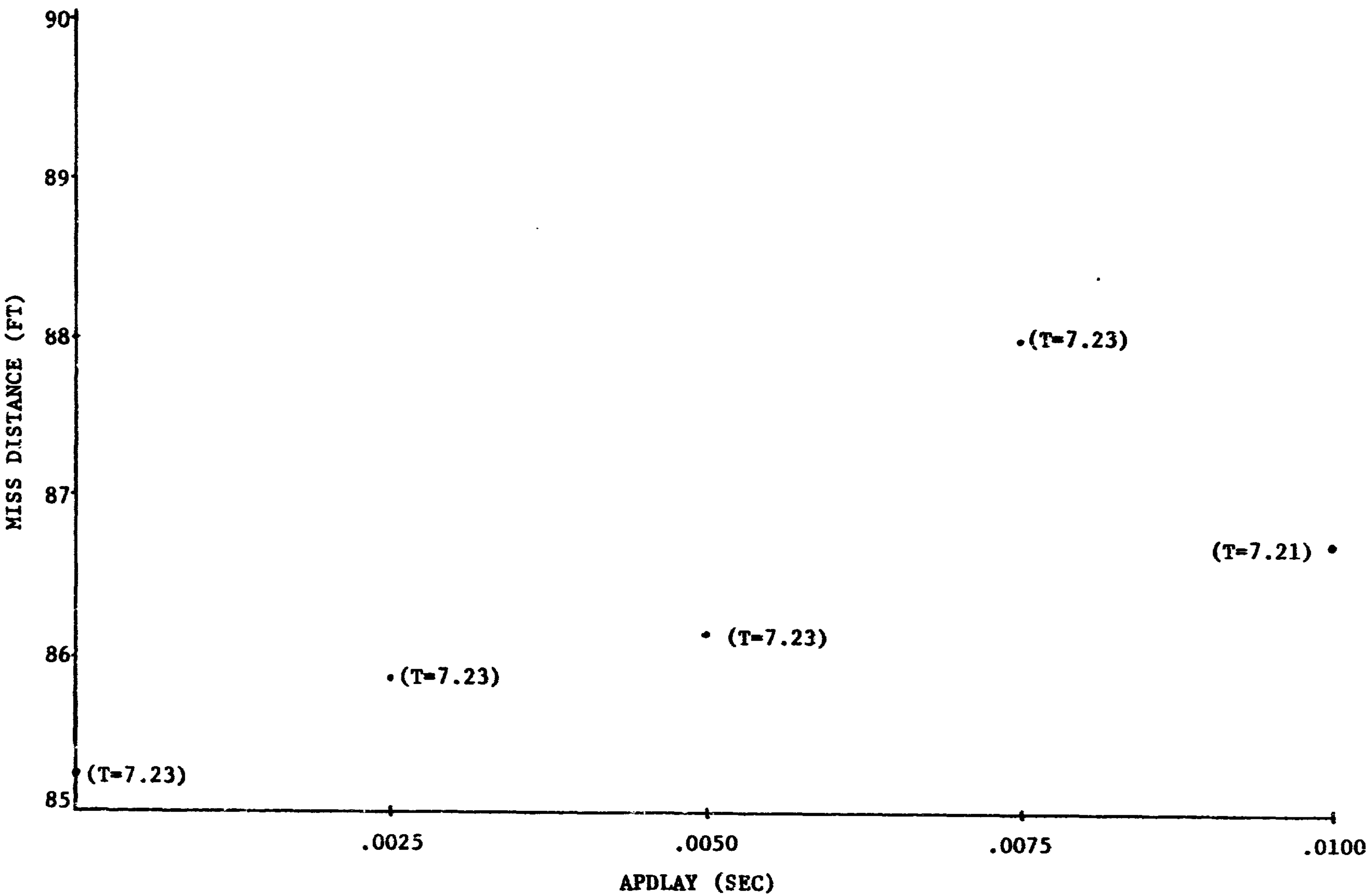


Figure D-21. Initial Heading Error, Climb/Dive Tgt, Stochastic, DT = .01 sec

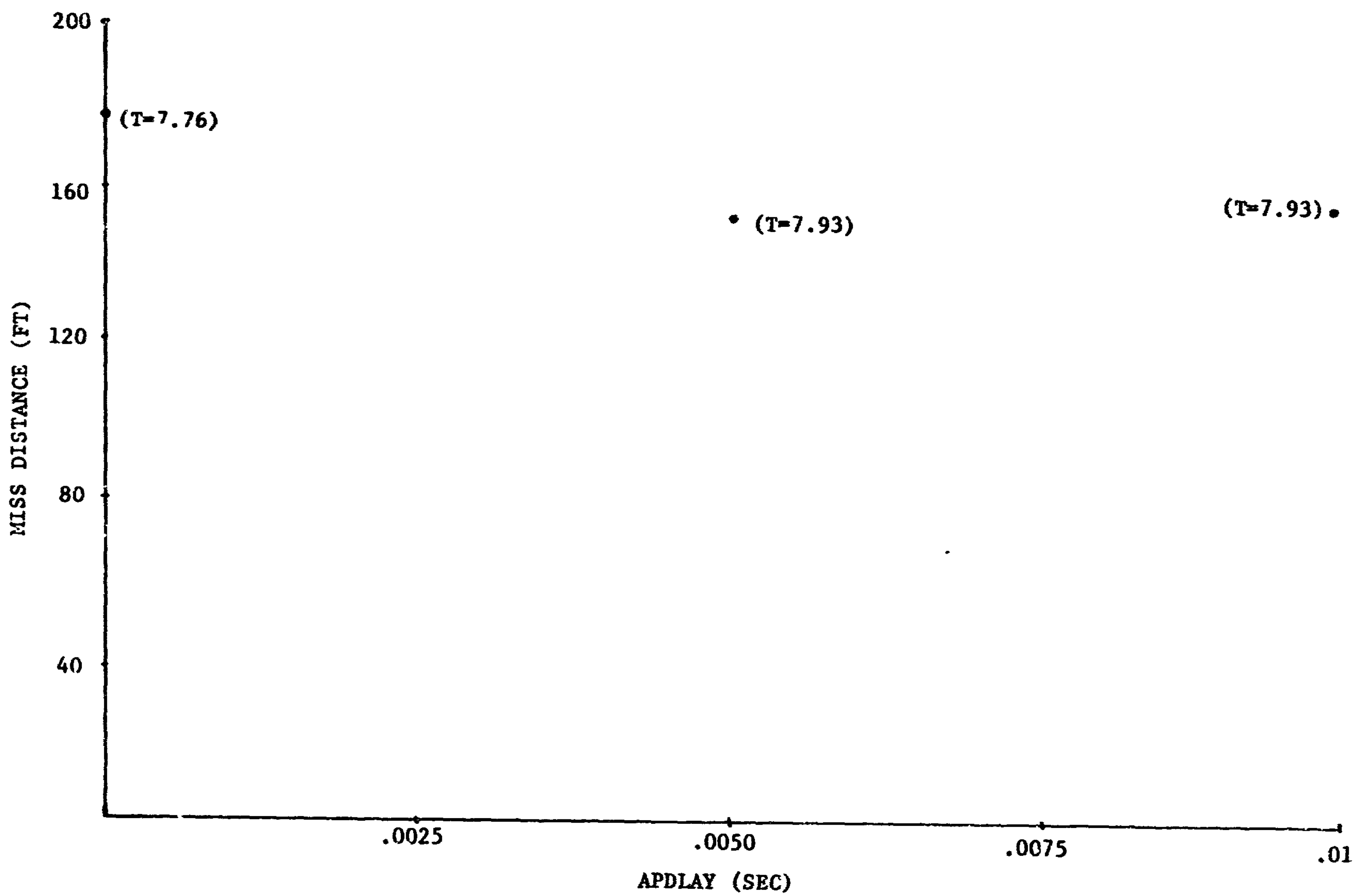


Figure D-22. Tail Attack, Str/Lvl Tgt, Stochastic, DT = .1 sec

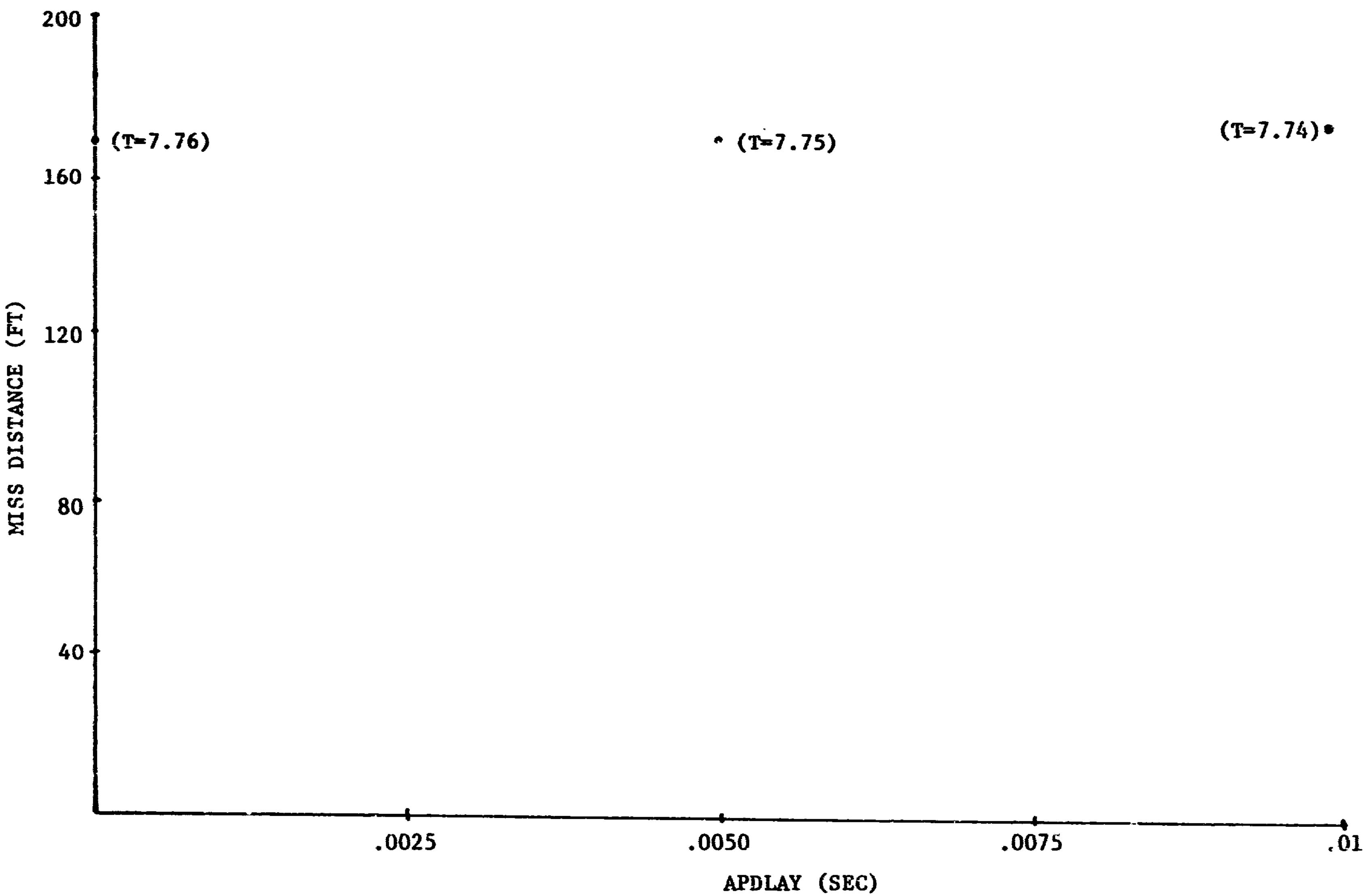


Figure D-23. Tail Attack, Turning Tgt, Stochastic, DT = .1 sec

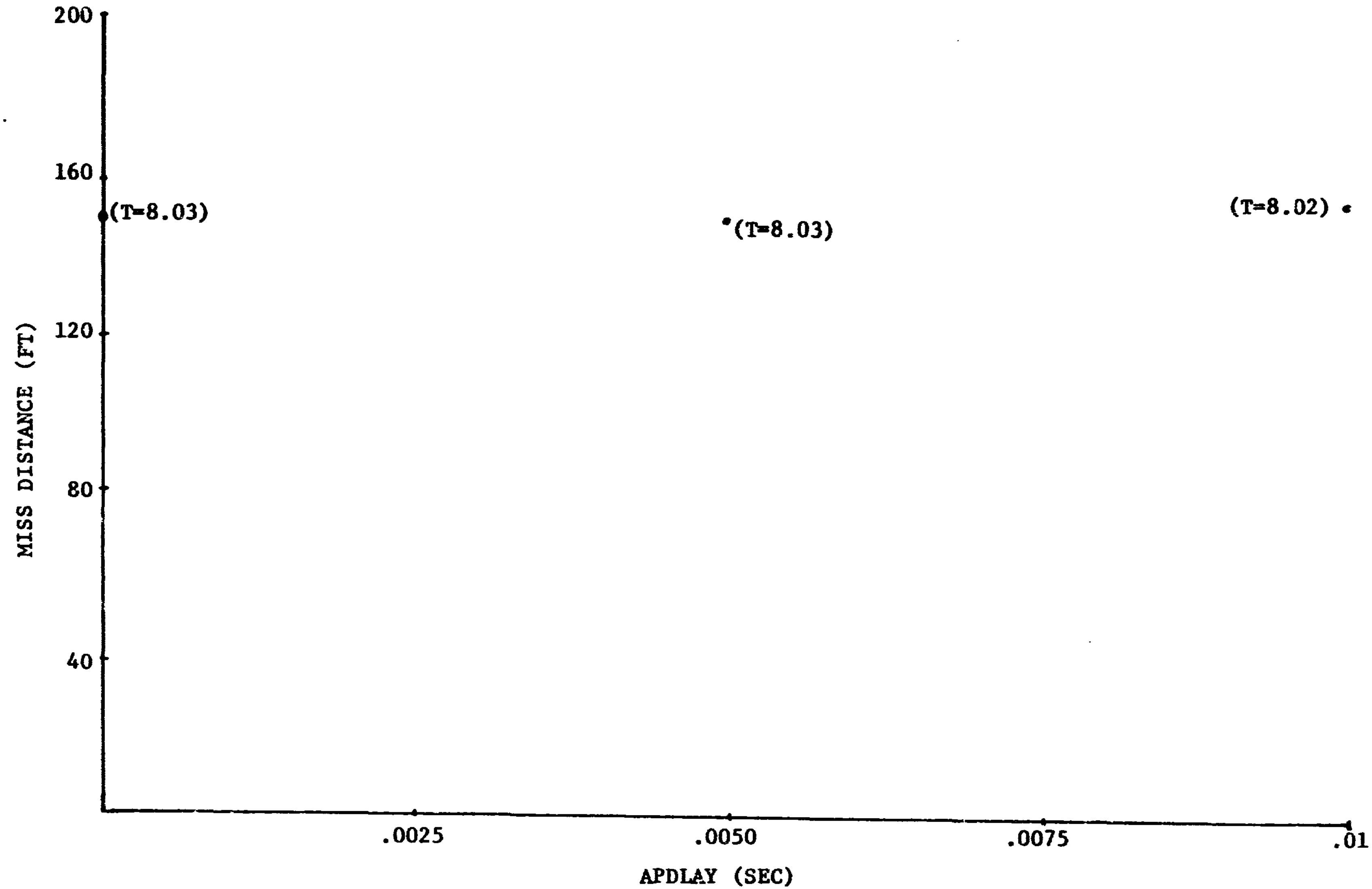


Figure D-24. Tail Attack, Climb/Dive Tgt, Stochastic, DT = .1 sec

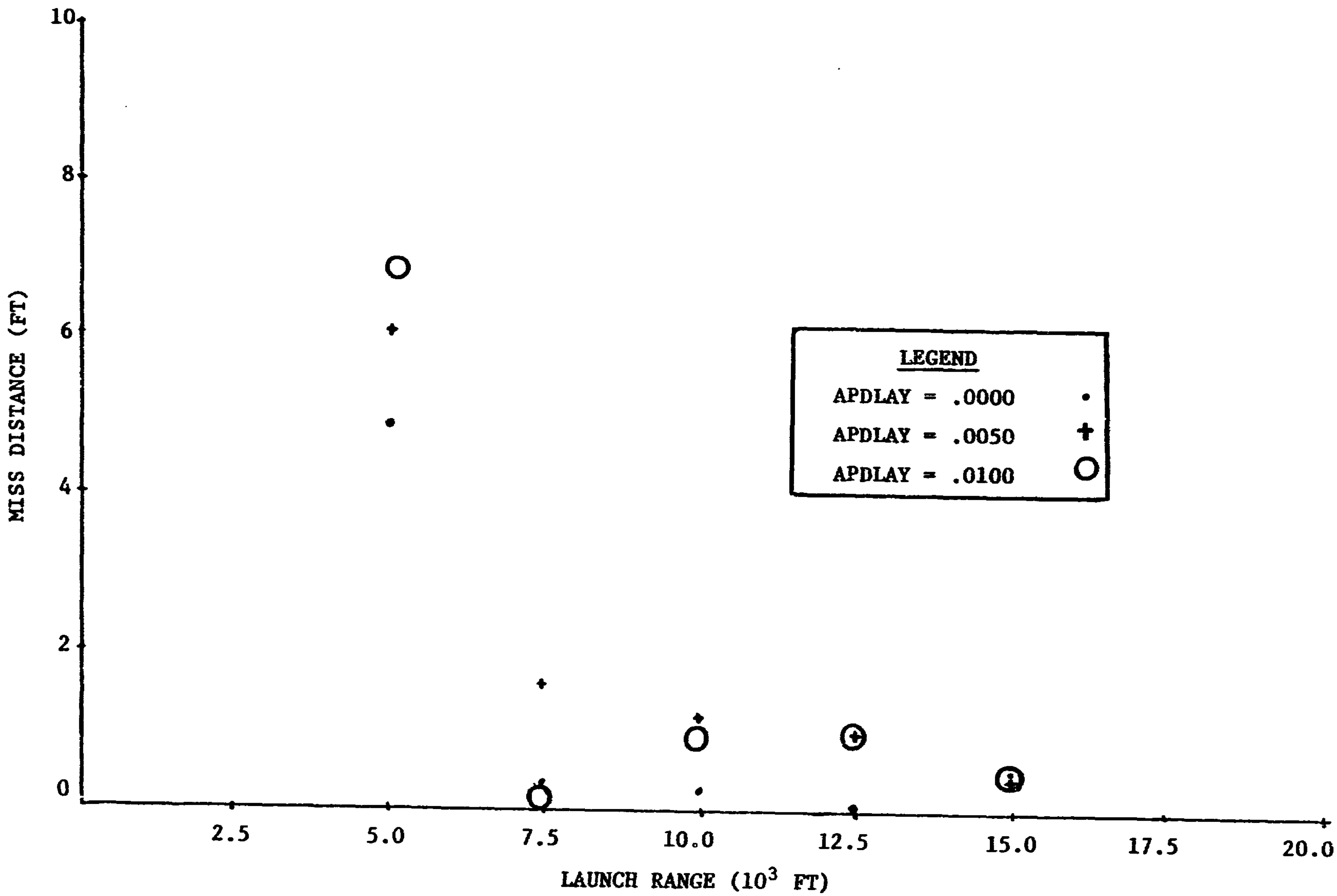


Figure D-25. Tail Attack, Str/Lvl Tgt, Variable Launch Range, Deterministic, DT = .01 sec

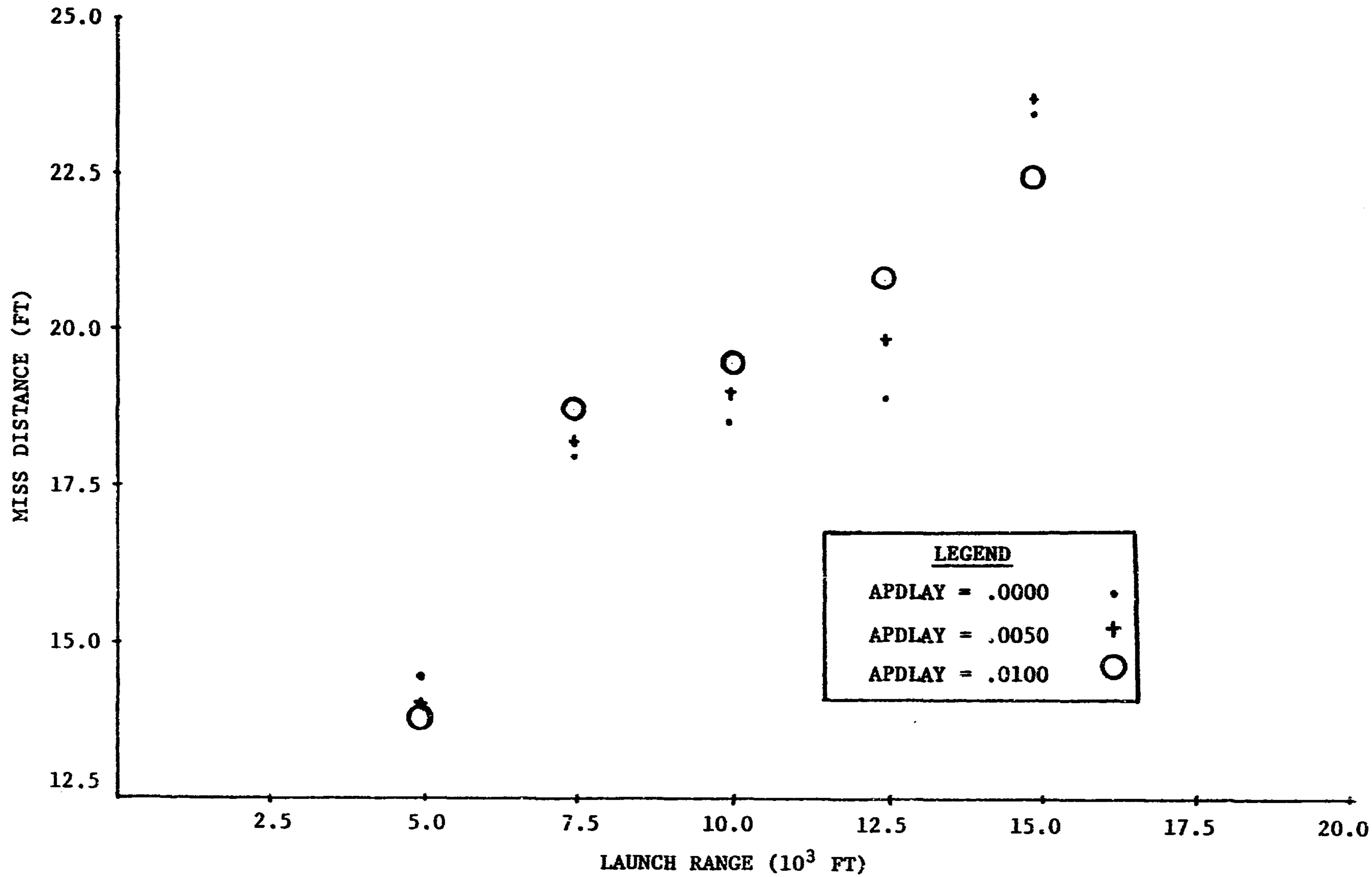


Figure D-26. Tail Attack, Turning Tgt, Variable Launch Range, Deterministic, DT = .01 sec

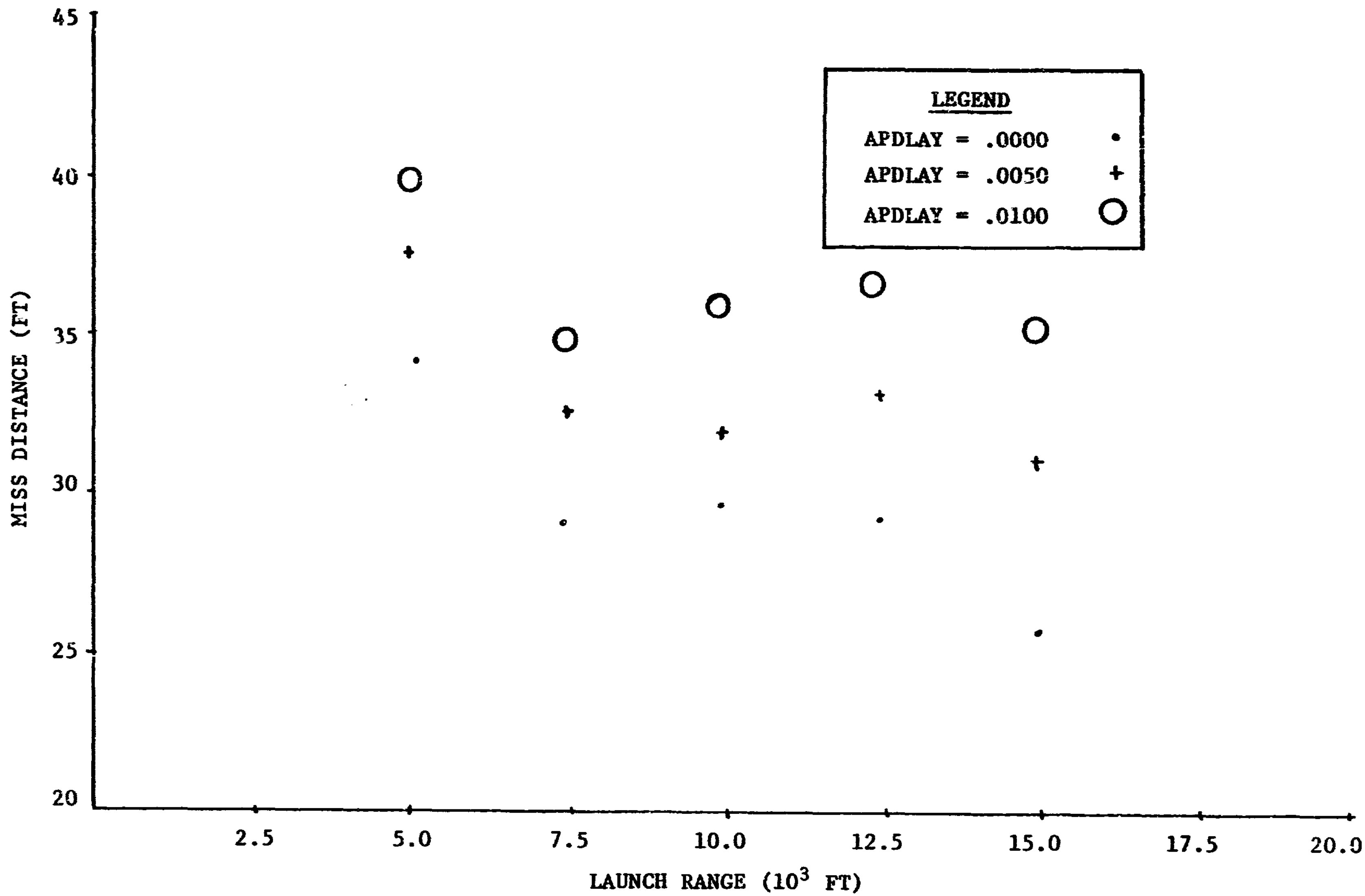
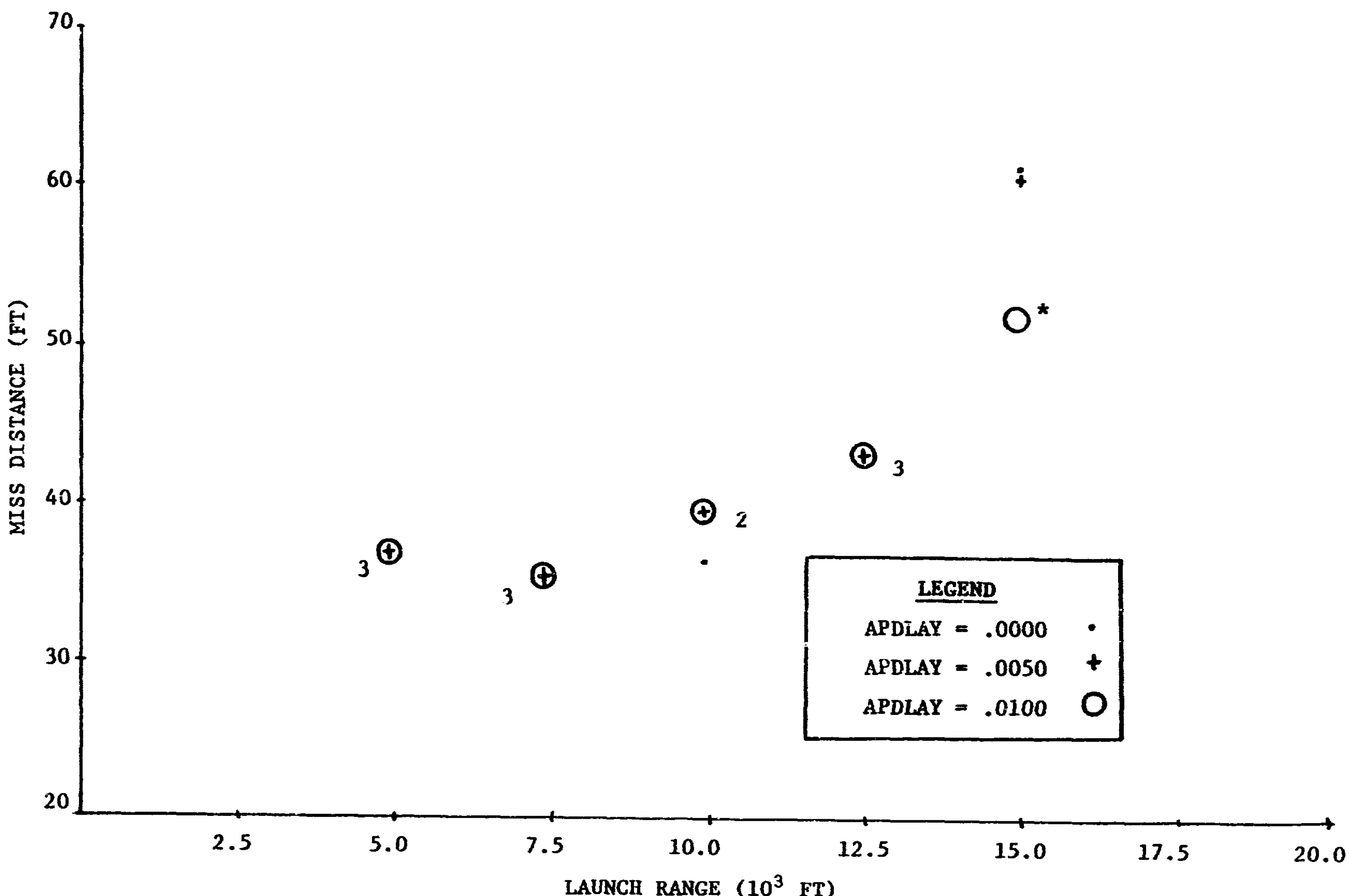


Figure D-27. Tain Attack, Climb/Dive Tgt, Variable Launch Range, Deterministic, DT = .01 sec



*Based on 16 Runs

Figure D-28. Tail Attack, Str/Lvl Tgt, Variable Launch Range, Stochastic, DT = .01 sec

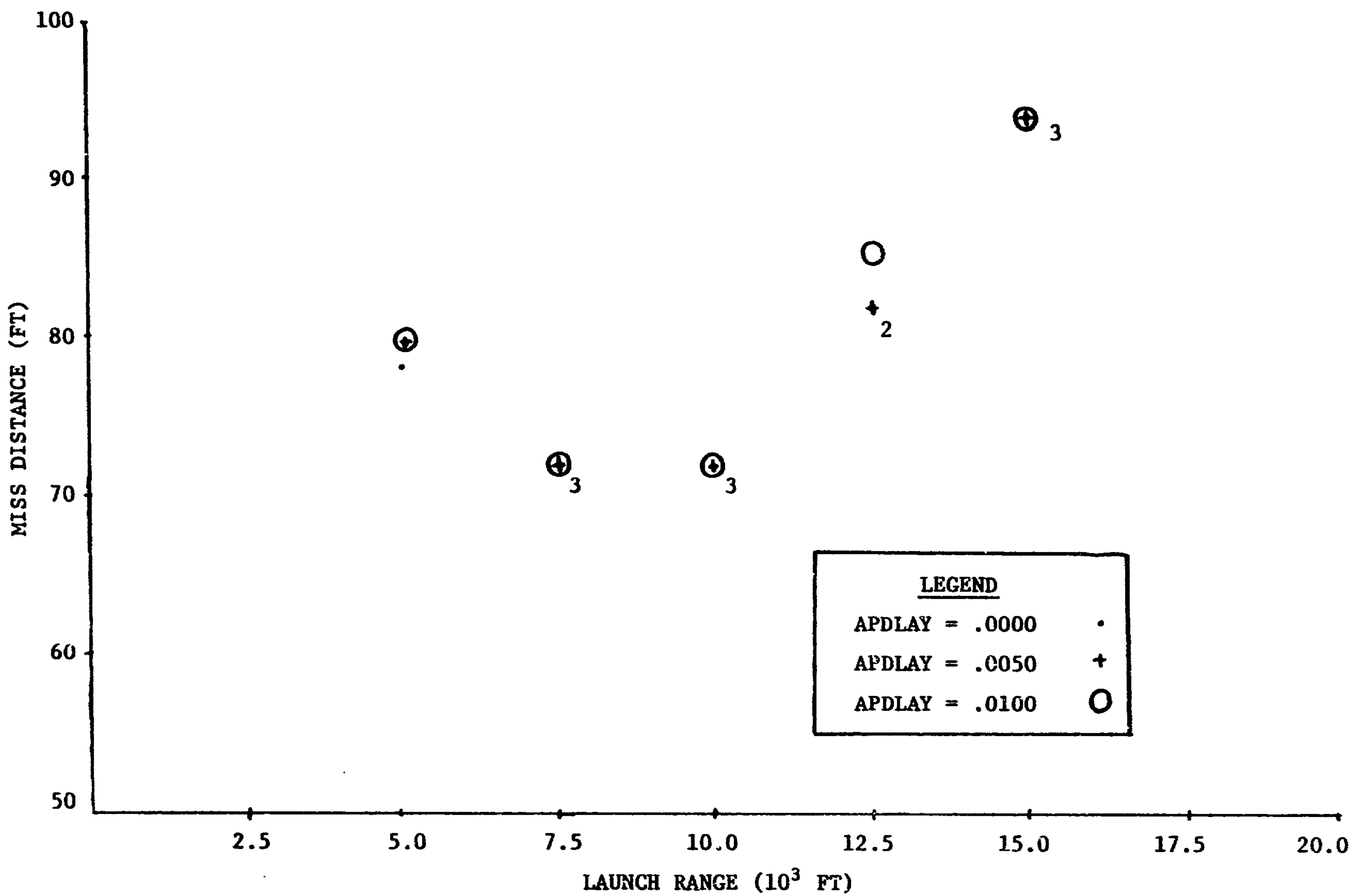


Figure D-29. Tail Attack, Turning Tgt, Variable, Launch Range, Stochastic, DT = .01 sec

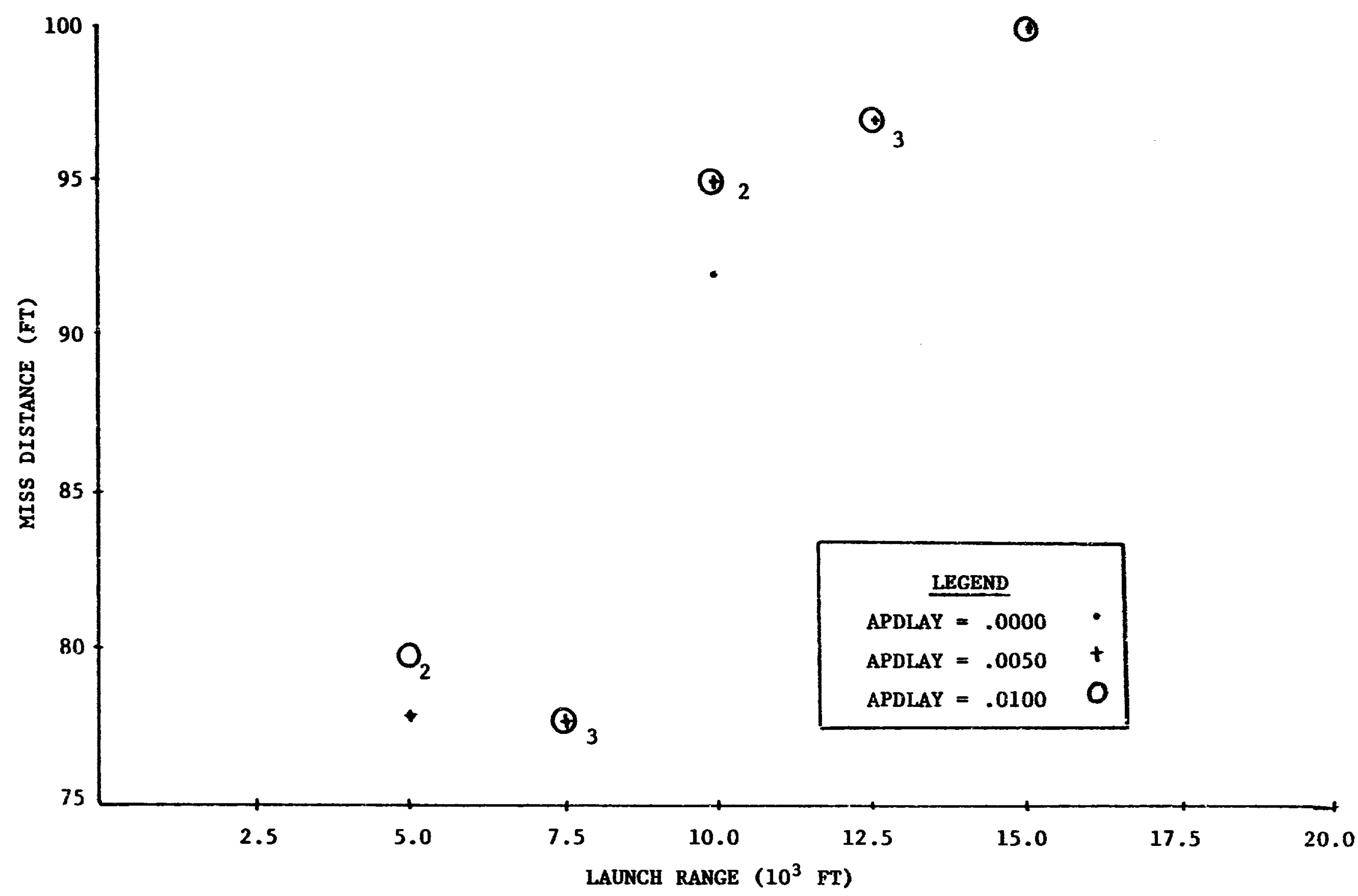


Figure D-30. Tail Attack, Climb/Dive Tgt, Variable Launch Range, Stochastic, DT = .01 sec

Appendix E

Postprocessor Plots

The Tactics IV postprocessor* produces plots of a number of engagement variables. These plots provide useful insights into the dynamics of the missile engagement. Altogether, 96 plots were generated for this study. Before they are introduced, however, a word is necessary about the type of plots produced and the test cases considered.

The four plots produced were XY versus time, horizontal acceleration (Acoma) versus time, vertical acceleration (Acomd) versus time and missile velocity versus time. These plots were chosen because they provide information about g limiting and energy loss. This data was used in the results tables of Chapter V.

Plots were produced for a limited number of test cases. These test cases considered only the tail attack geometry. Also, Apdlay was set to zero for all of the runs. This was safely done as even maximum Apdlay produced indiscernible differences on the plots. Next, both the fixed launch range (10Kft) and variable launch ranges (5Kft and 15Kft) were used. Additionally, both deterministic and stochastic missile models were used. Next, the basic integration step size (DT) was set to both .01 seconds and .1 seconds for the 10Kft launch range tests. A DT of .01 seconds was used for the variable launch range tests. Finally, the straight and level (Str/lvl), turning and climb/dive target evasive maneuvers were considered. These variables are listed along with their respective figure numbers in Table E-I.

Table E-I

Postprocessor Plots

Launch Range	Missile Model	DT	Target Maneuver	Figure Number
10Kft	Deterministic	.01 sec	Str/Lvl	E-1 - E-4
10Kft	Deterministic	.01 sec	Turning	E-5 - E-8
10Kft	Deterministic	.01 sec	Climb/ Dive	E-9 - E-12
10Kft	Deterministic	.1 sec	Str/Lvl	E-13 - E-16
10Kft	Deterministic	.1 sec	Turning	E-17 - E-20
10Kft	Deterministic	.1 sec	Climb/ Dive	E-21 - E-24
10Kft	Stochastic	.01 sec	Str/Lvl	E-25 - E-28
10Kft	Stochastic	.01 sec	Turning	E-29 - E-32
10Kft	Stochastic	.01 sec	Climb/ Dive	E-33 - E-36
10Kft	Stochastic	.1 sec	Str/Lvl	E-37 - E-40
10Kft	Stochastic	.1 sec	Turning	E-41 - E-44
10Kft	Stochastic	.1 sec	Climb/ Dive	E-45 - E-48
5Kft	Deterministic	.01 sec	Str/Lvl	E-49 - E-52
5Kft	Deterministic	.01 sec	Turning	E-53 - E-56
5Kft	Deterministic	.01 sec	Climb/ Dive	E-57 - E-60
15Kft	Deterministic	.01 sec	Str/Lvl	E-61 - E-64
15Kft	Deterministic	.01 sec	Turning	E-65 - E-68
15Kft	Deterministic	.01 sec	Climb/ Dive	E-69 - E-72

Note: All plots were of the tail attack geometry with Apdlay set to zero.

Table E-I (Cont'd)

Postprocessor Plots

Launch Range	Missile Model	DT	Target Maneuver	Figure Number
5Kft	Stochastic	.01 sec	Str/Lvl	E-73 - E-76
5Kft	Stochastic	.01 sec	Turning	E-77 - E-80
5Kft	Stochastic	.01 sec	Climb/ Dive	E-81 - E-84
15Kft	Stochastic	.01 sec	Str/Lvl	E-85 - E-88
15Kft	Stochastic	.01 sec	Turning	E-89 - E-92
15Kft	Stochastic	.01 sec	Climb/ Dive	E-93 - E-96

Note: All plots were of the tail attack geometry with Apdlay set to zero.

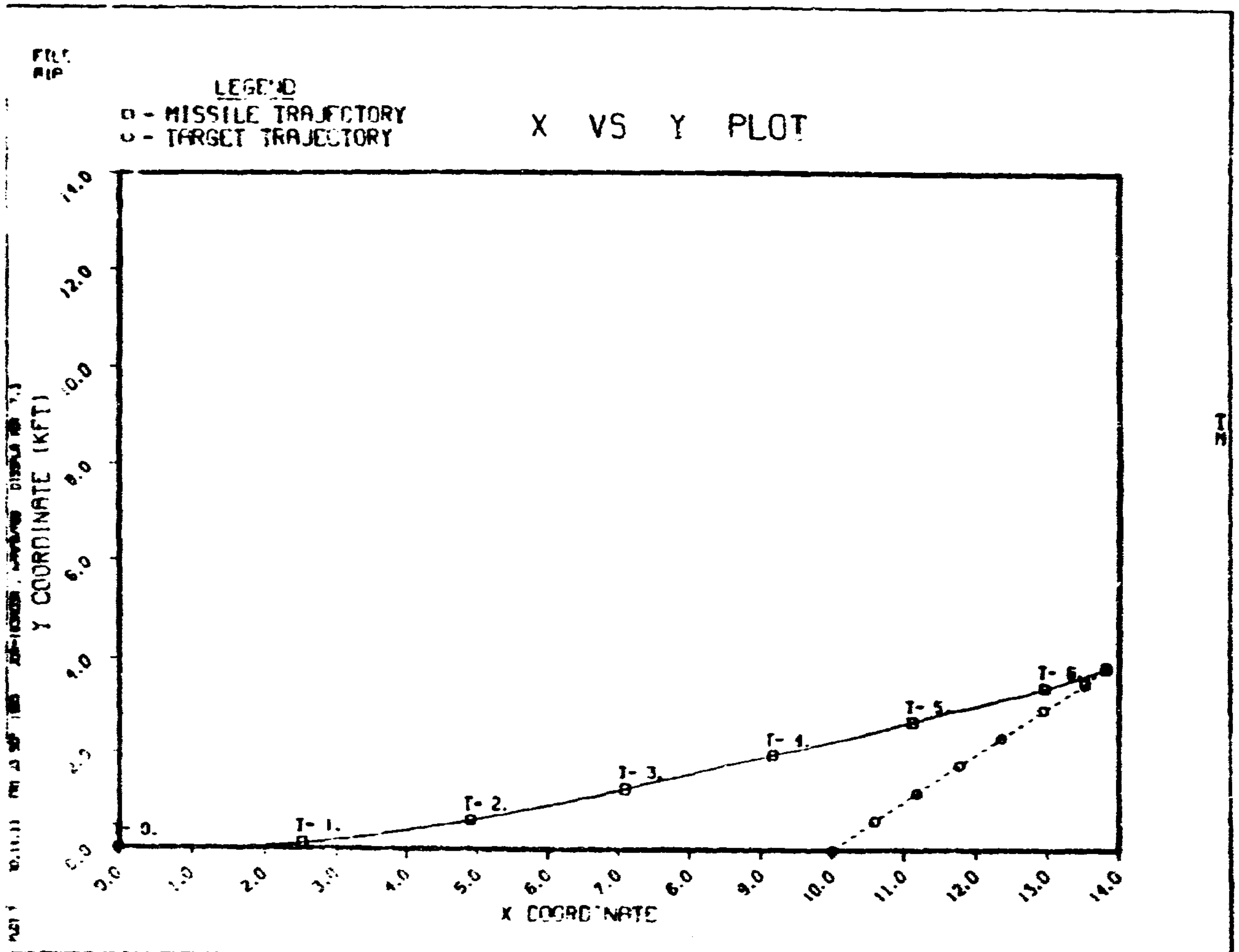


Figure E-1. X Y, Tail Attack, Str/Lvl Tgt, Deterministic, DT = .01 sec

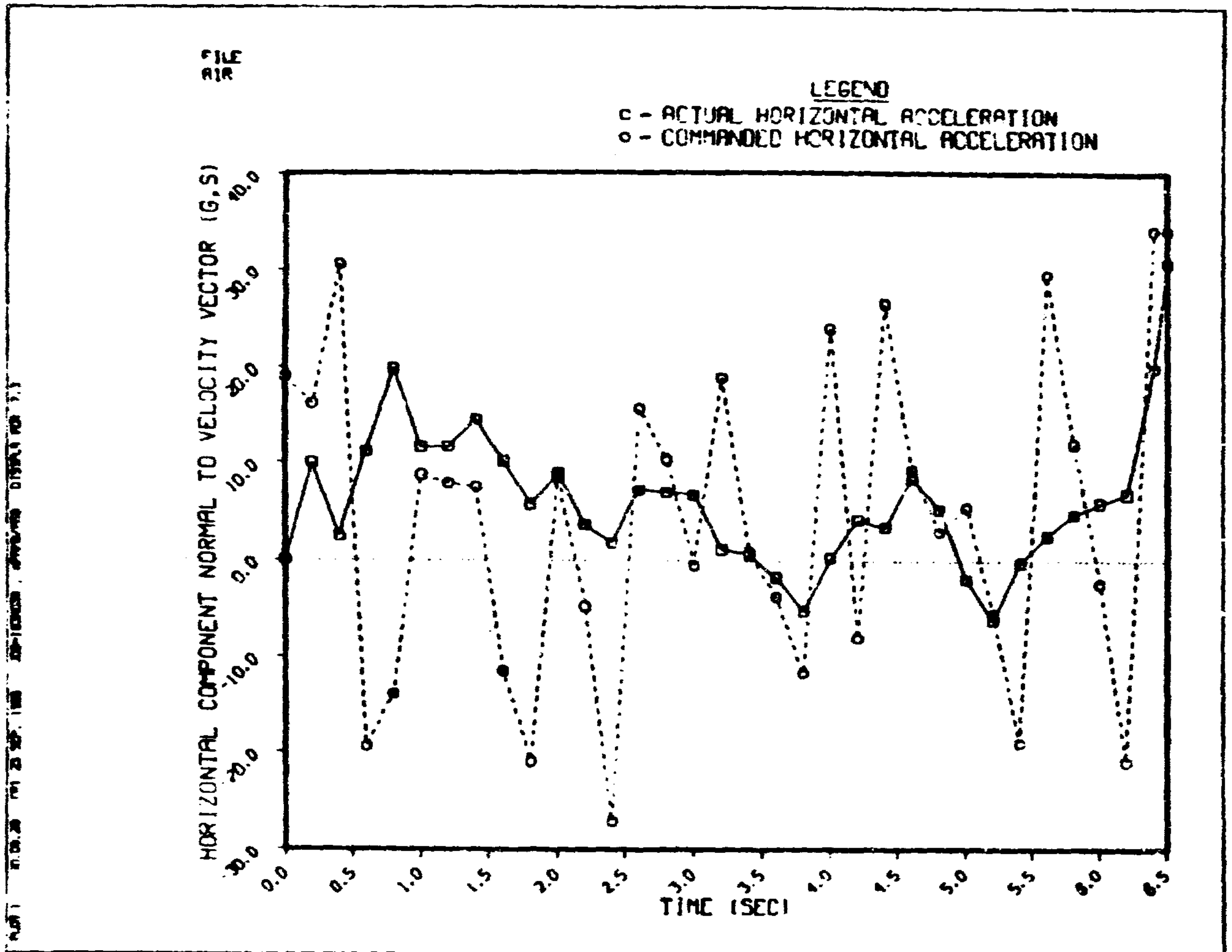


Figure E-2. Acoma, Tail Attack, Str/Lvl Tgt, Deterministic, DT = .01 sec

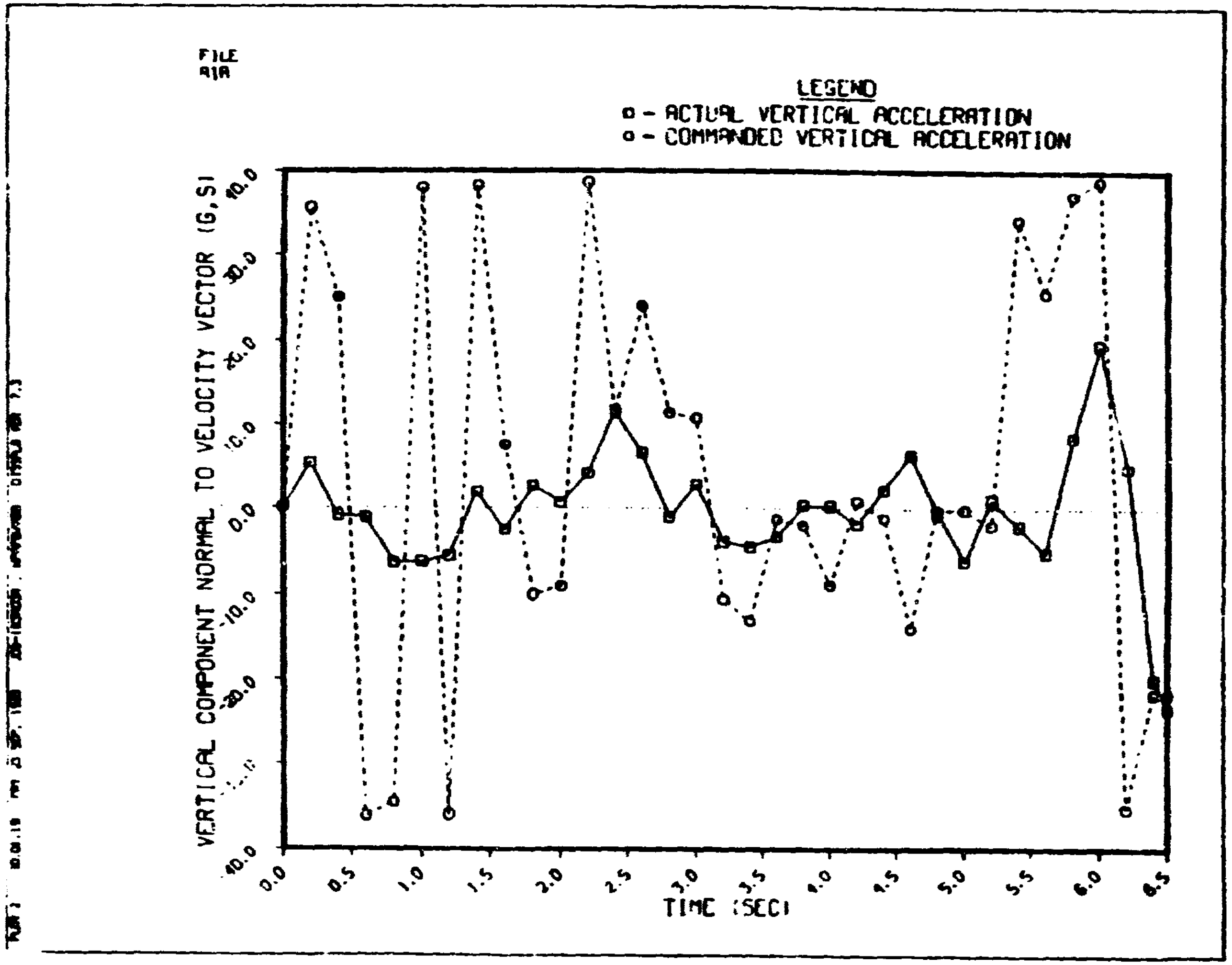


Figure E-3. Acomd, Tail Attack, Str/Lvl Tgt, Deterministic, DT = .01 sec

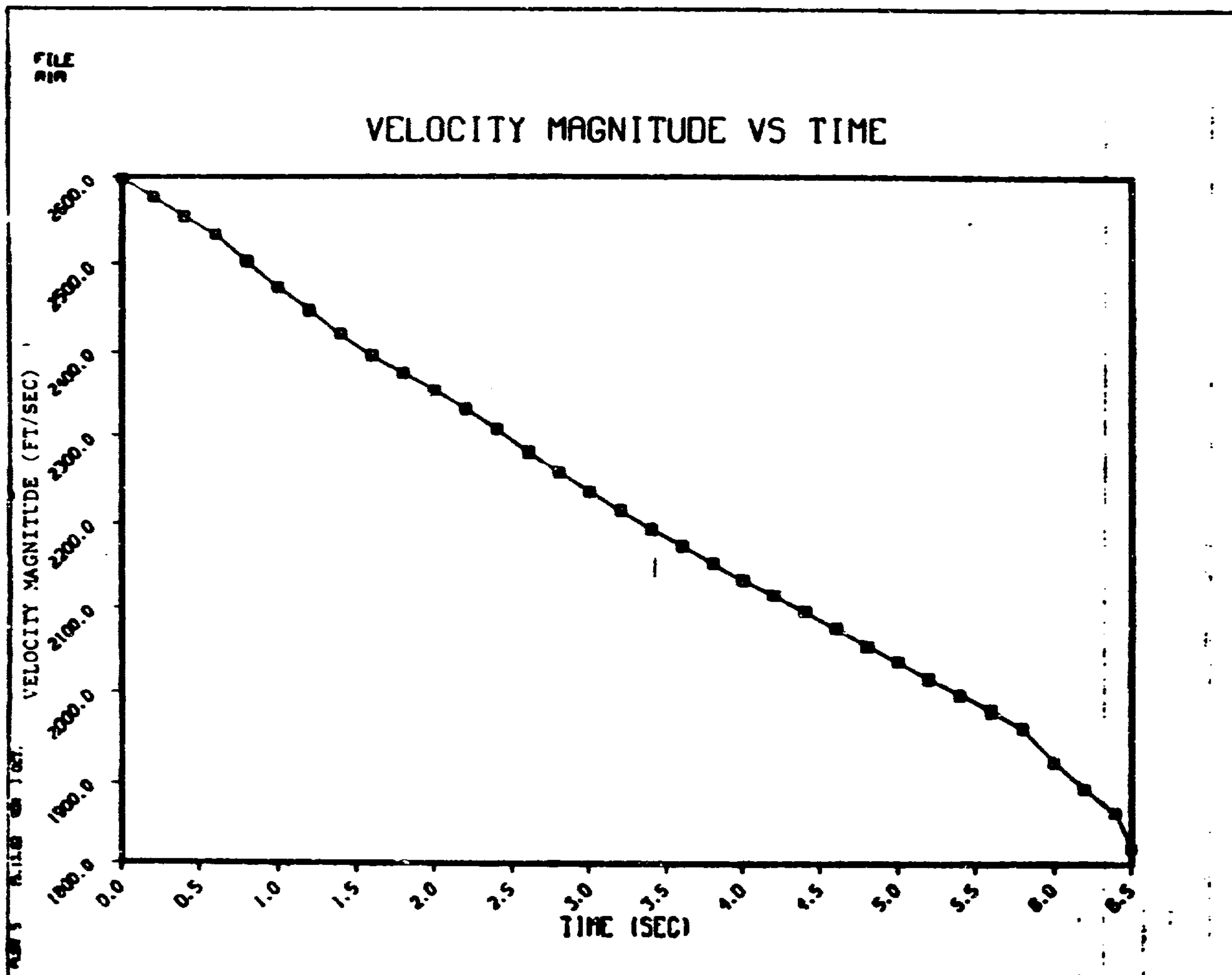


Figure E-4. Velocity vs Time, Str/Lvl, Deterministic, DT = .01 sec

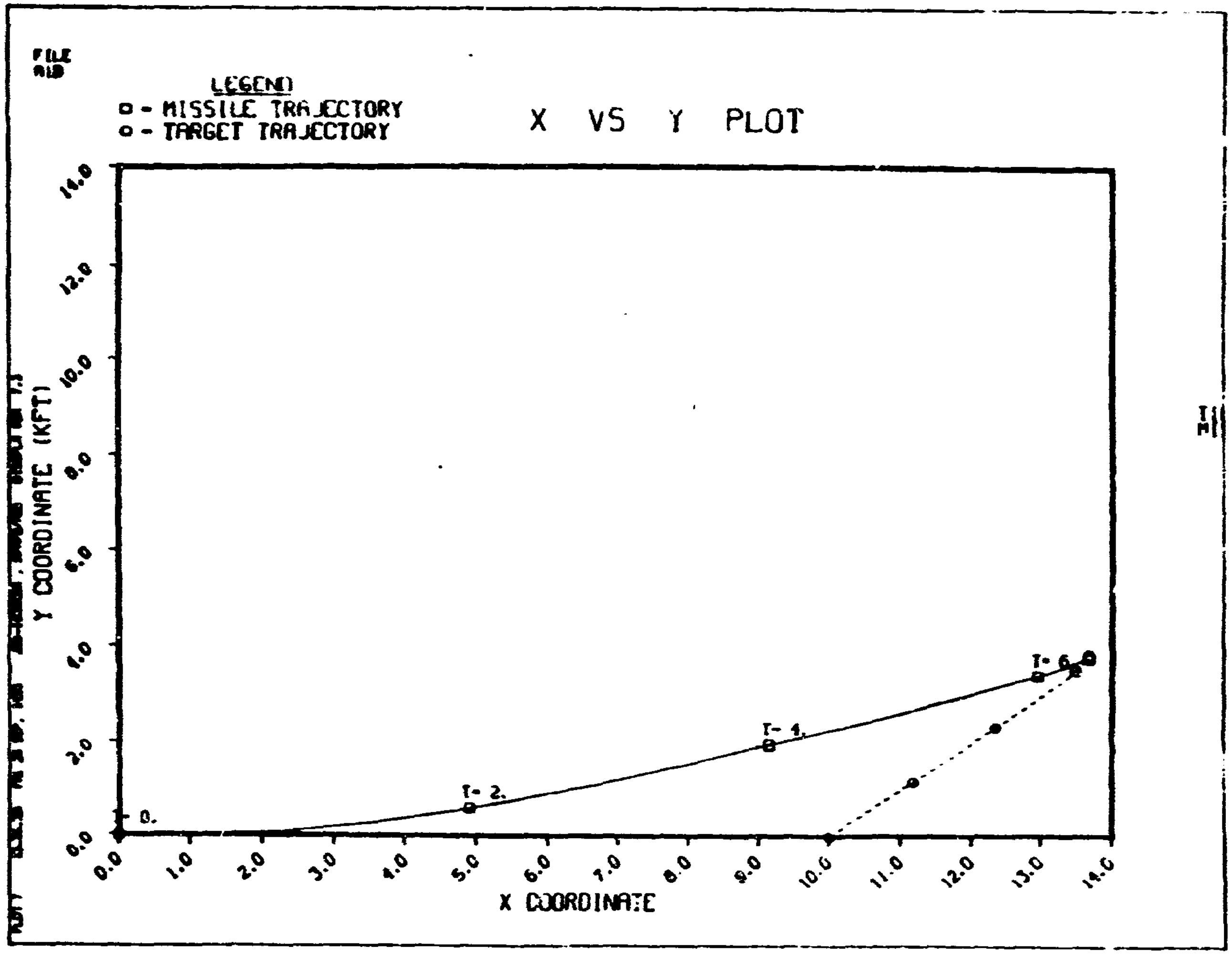


Figure E-5. X Y, Tail Attack, Turning Tgt, Deterministic, DT = .01 sec

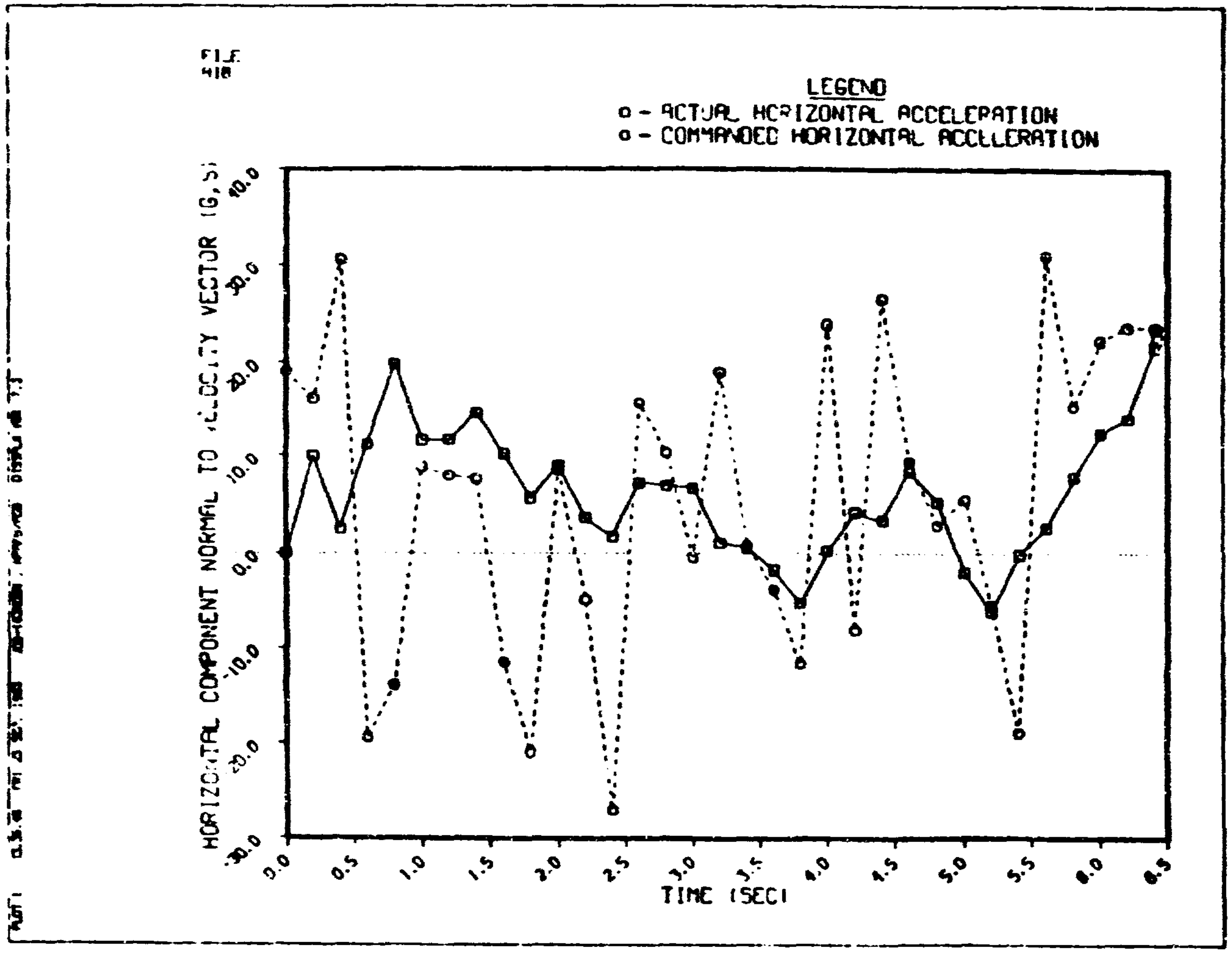


Figure E-6. Acoma, Tail Attack, Turning Tgt, Deterministic, DT = .01 sec

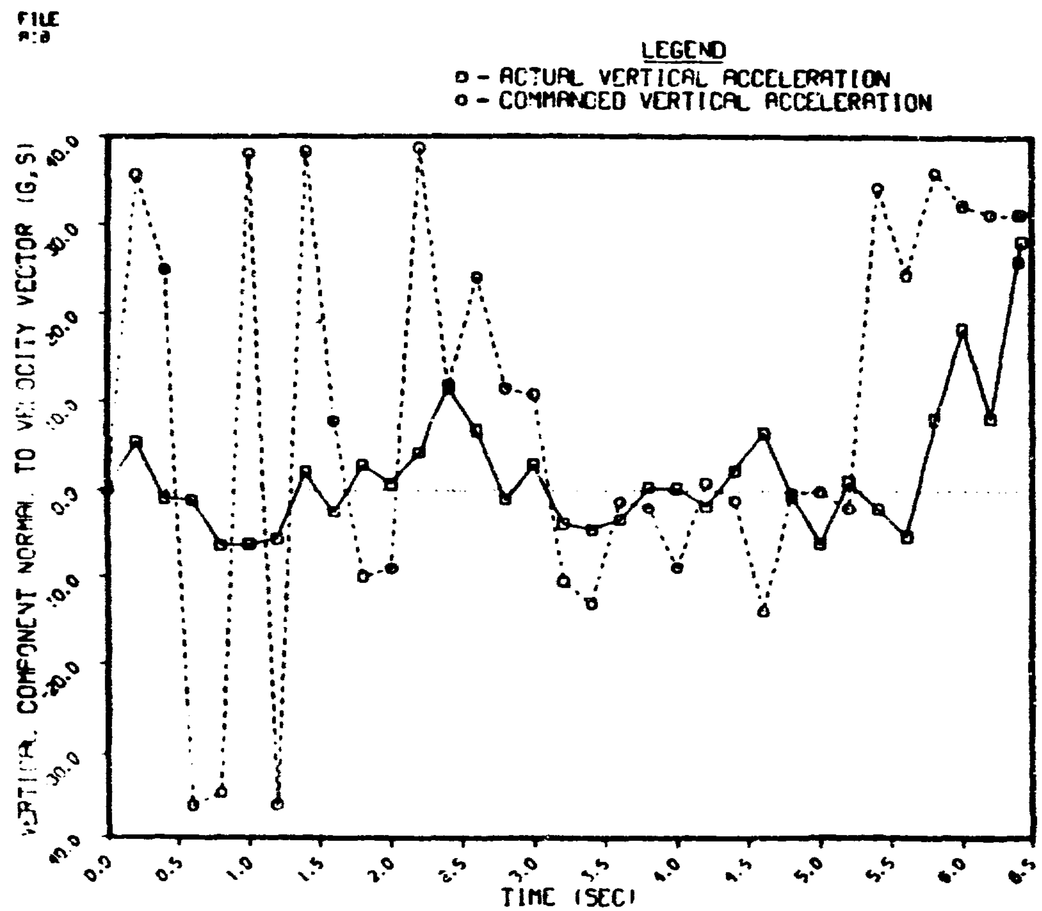


Figure E-7. Acomd, Tail Attack, Turning Tgt, Deterministic, DT = .01 sec

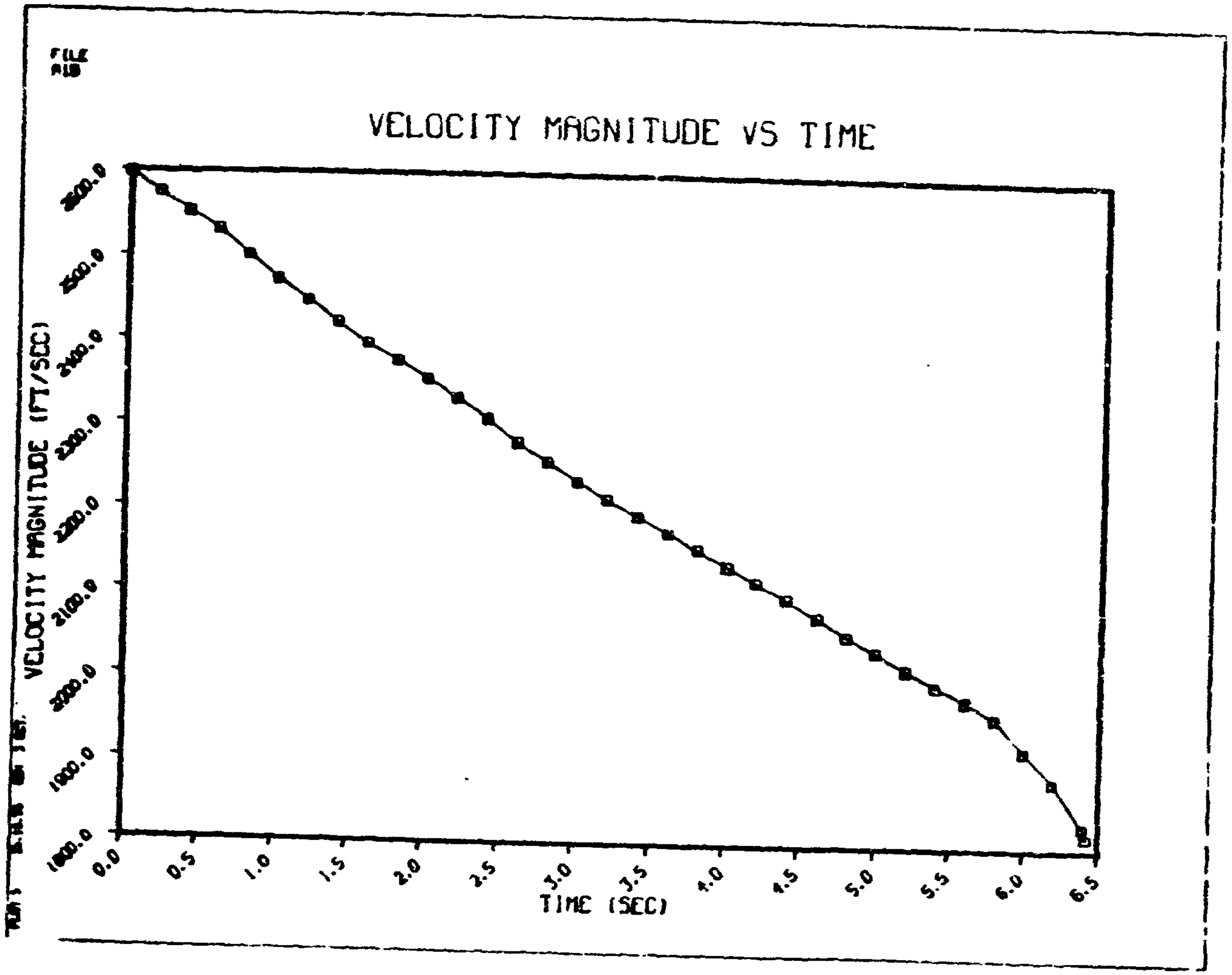


Figure E-8. Velocity vs Time, Tail Attack, Turning Tgt, Deterministic, DT = .01 sec

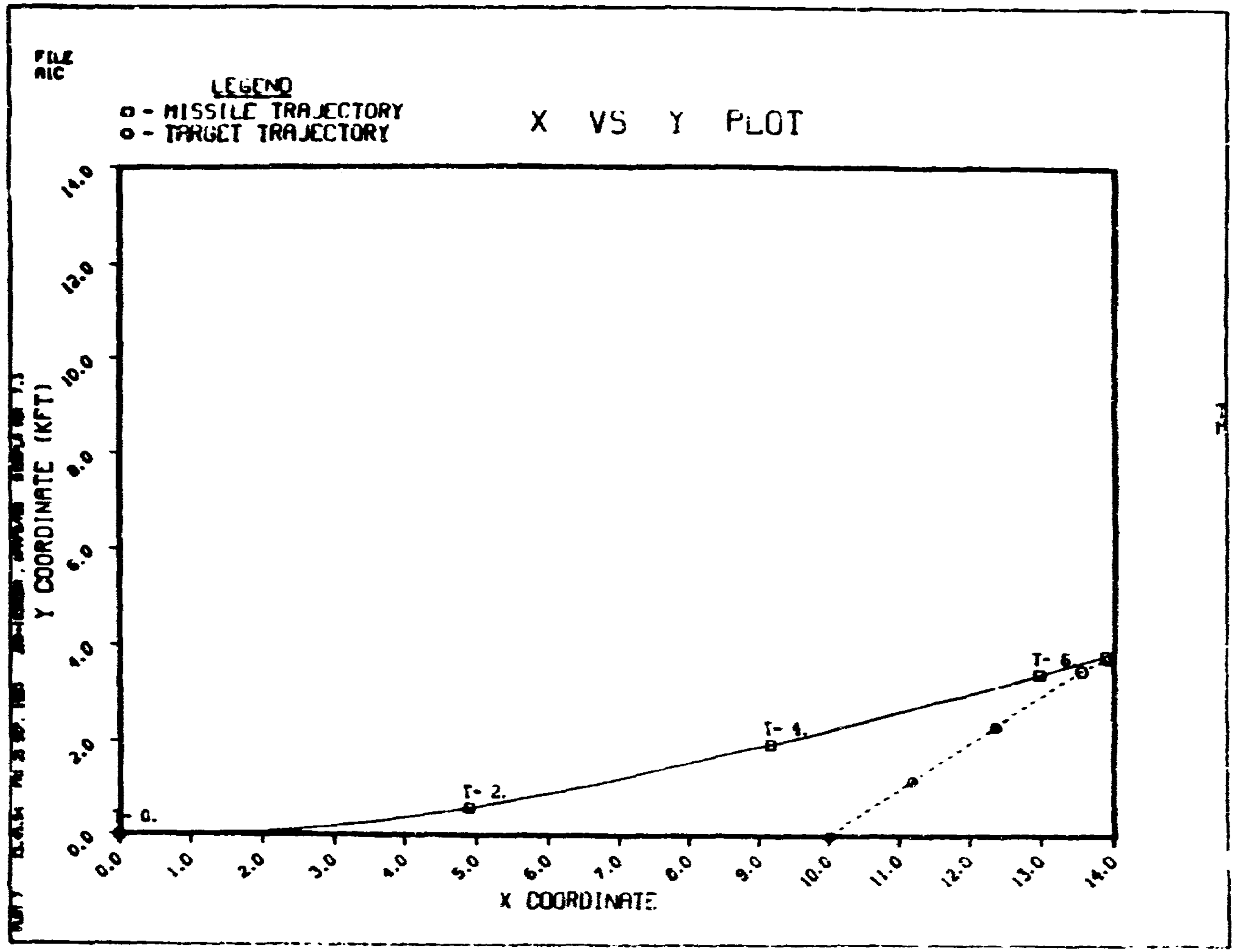


Figure E-9. X Y, Tail Attack, Climb/Dive Tgt, Deterministic, DT = .01 sec

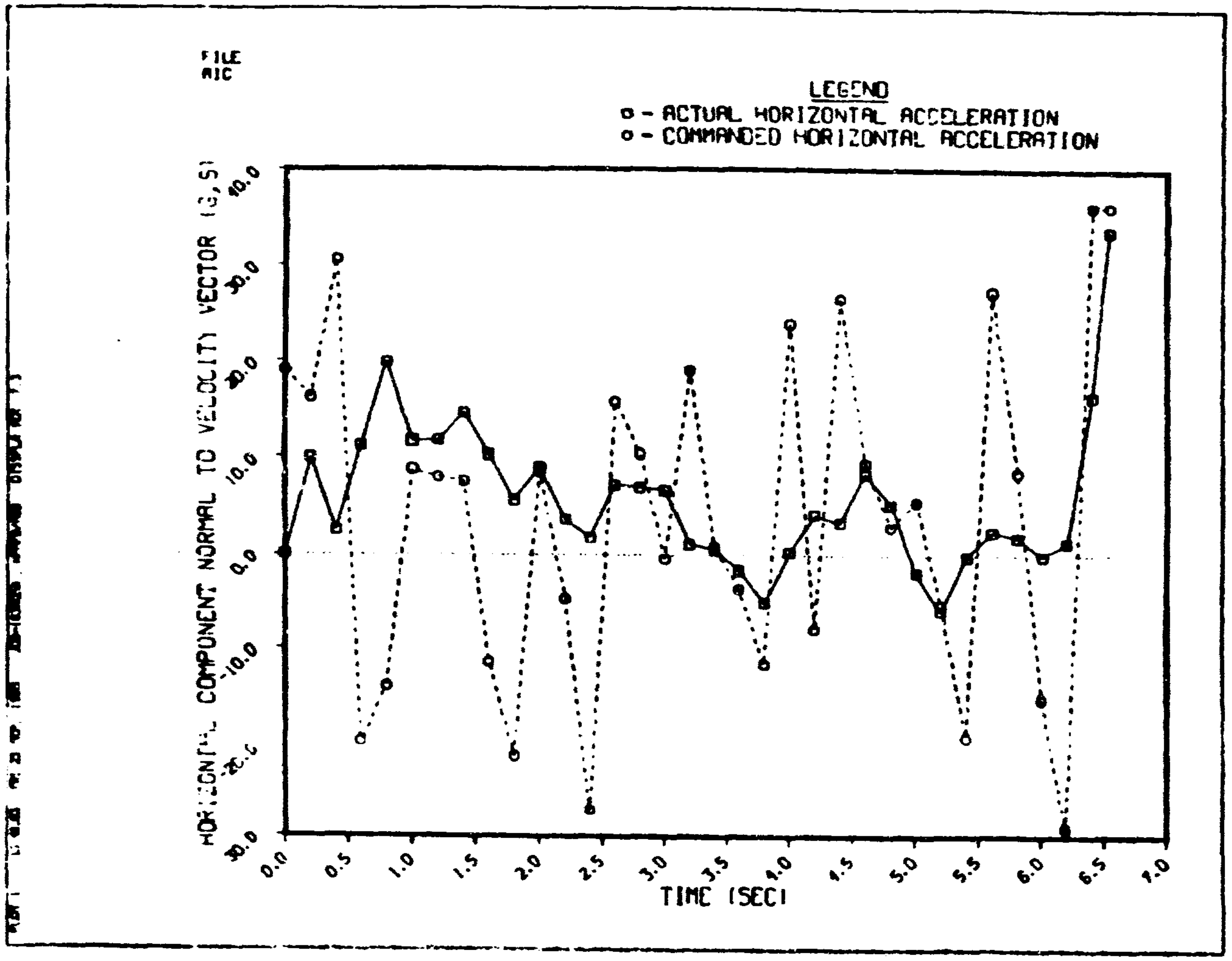


Figure E-10. Acoma, Tail Attack, Climb/Dive Tgt, Deterministic, DT = .01 sec

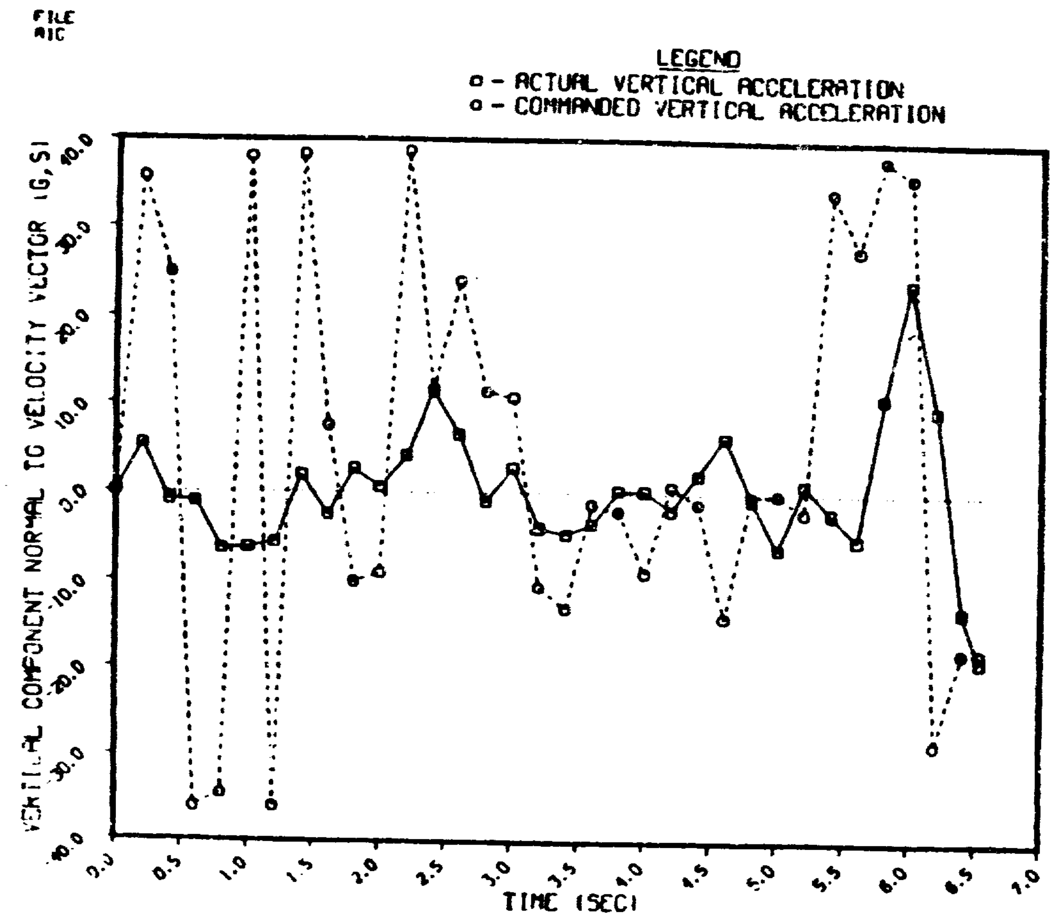


Figure E-11. Acomd, Tail Attack, Climb/Dive Tgt, Deterministic, DT = .01 sec

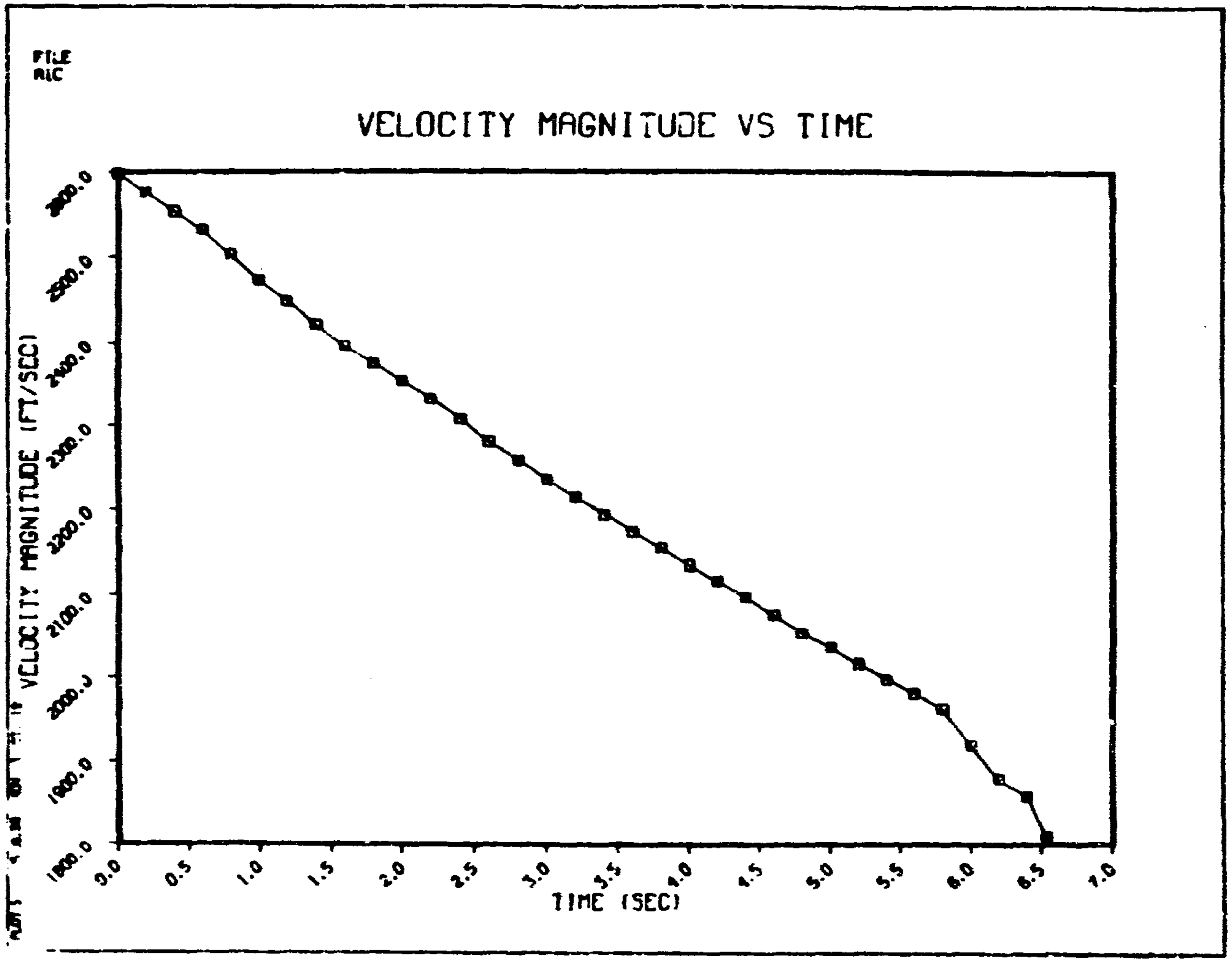


Figure E-12. Velocity vs Time, Tail Attack, Climb/Dive Tgt, Deterministic, DT = .01 sec

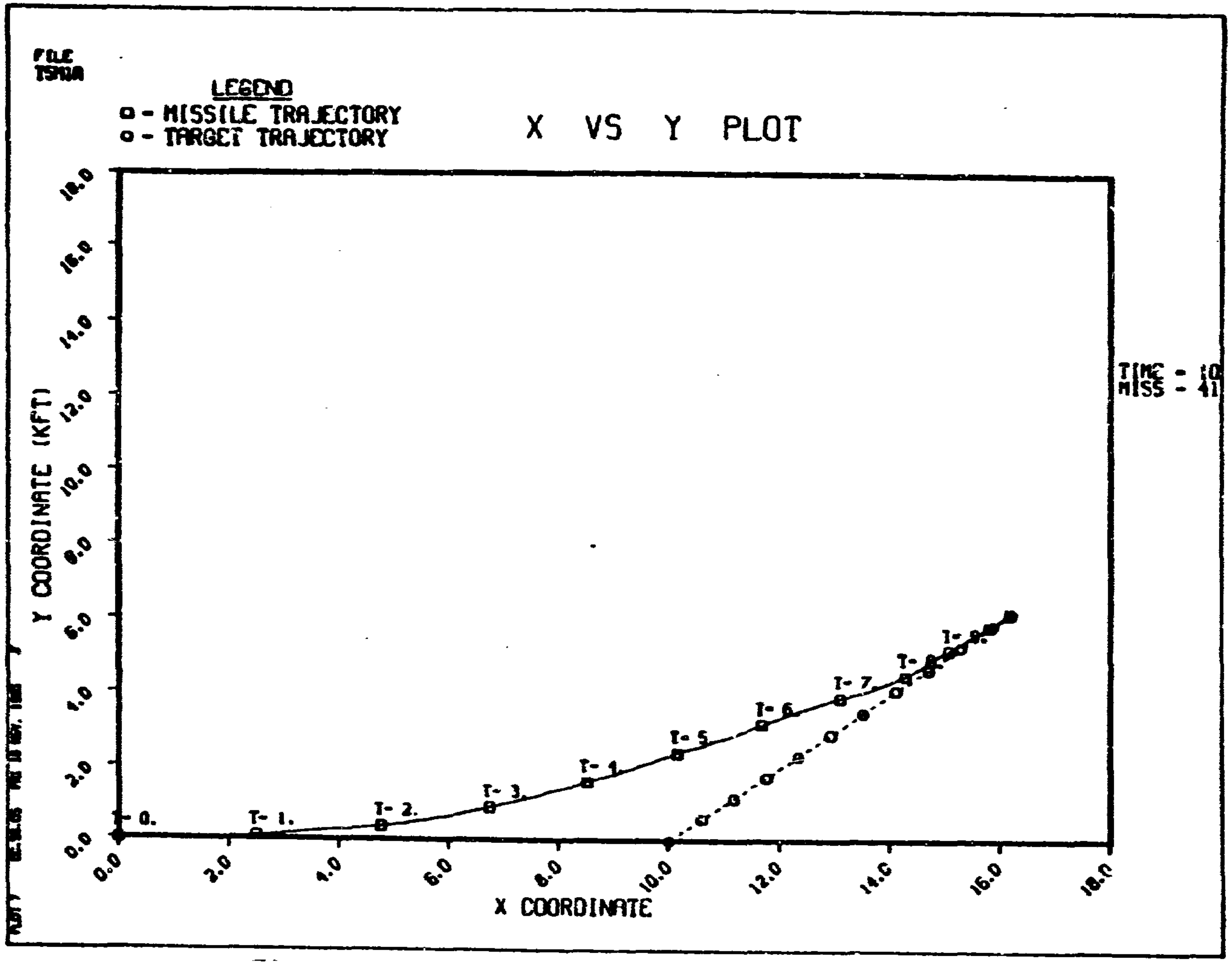


Figure E-13. X Y, Tail Attack, Str/Lvl Tgt, Deterministic, DT = .1 sec

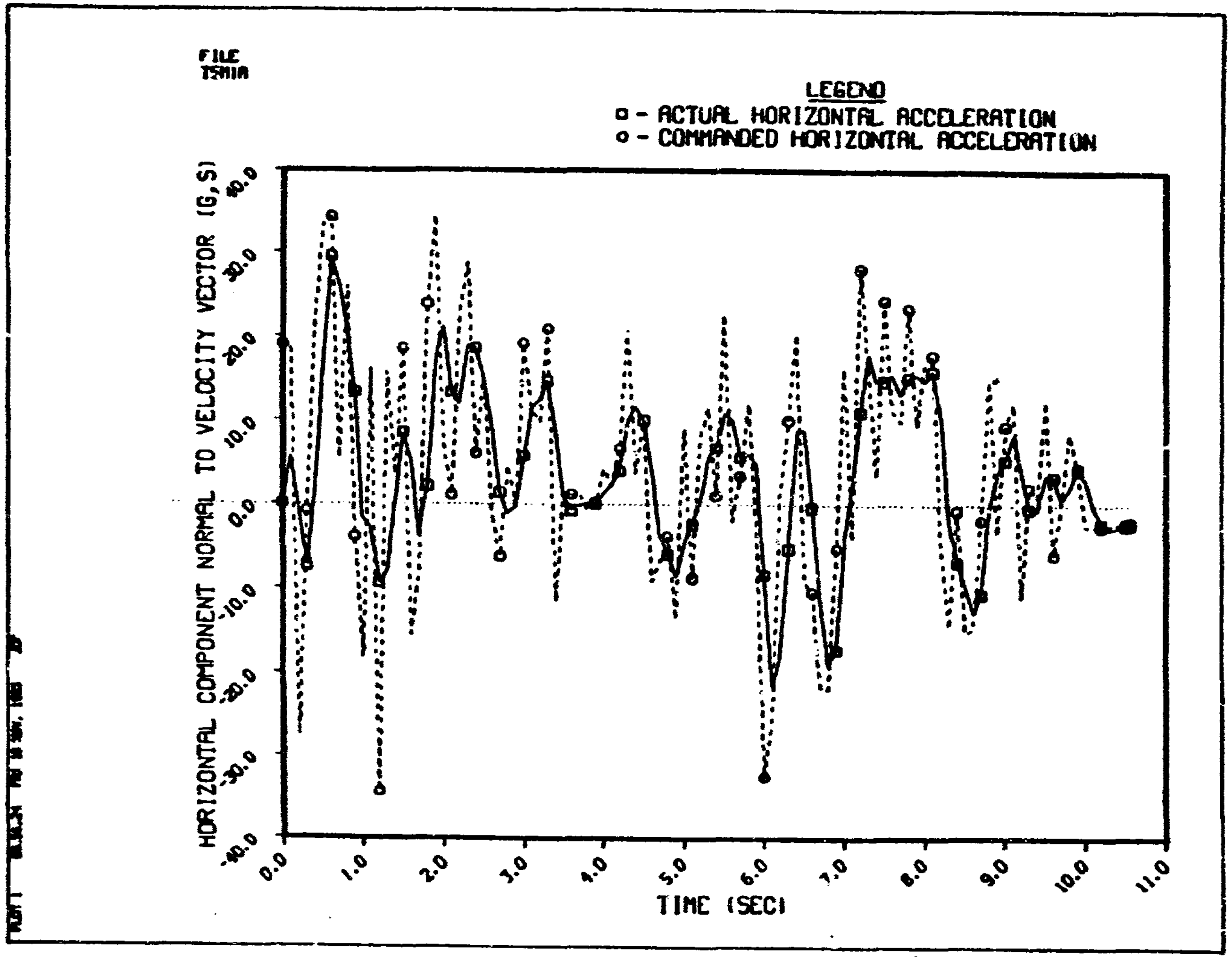


Figure E-14. Acoma, Tail Attack, Str/Lvl Tgt, Deterministic, DT = .1 sec

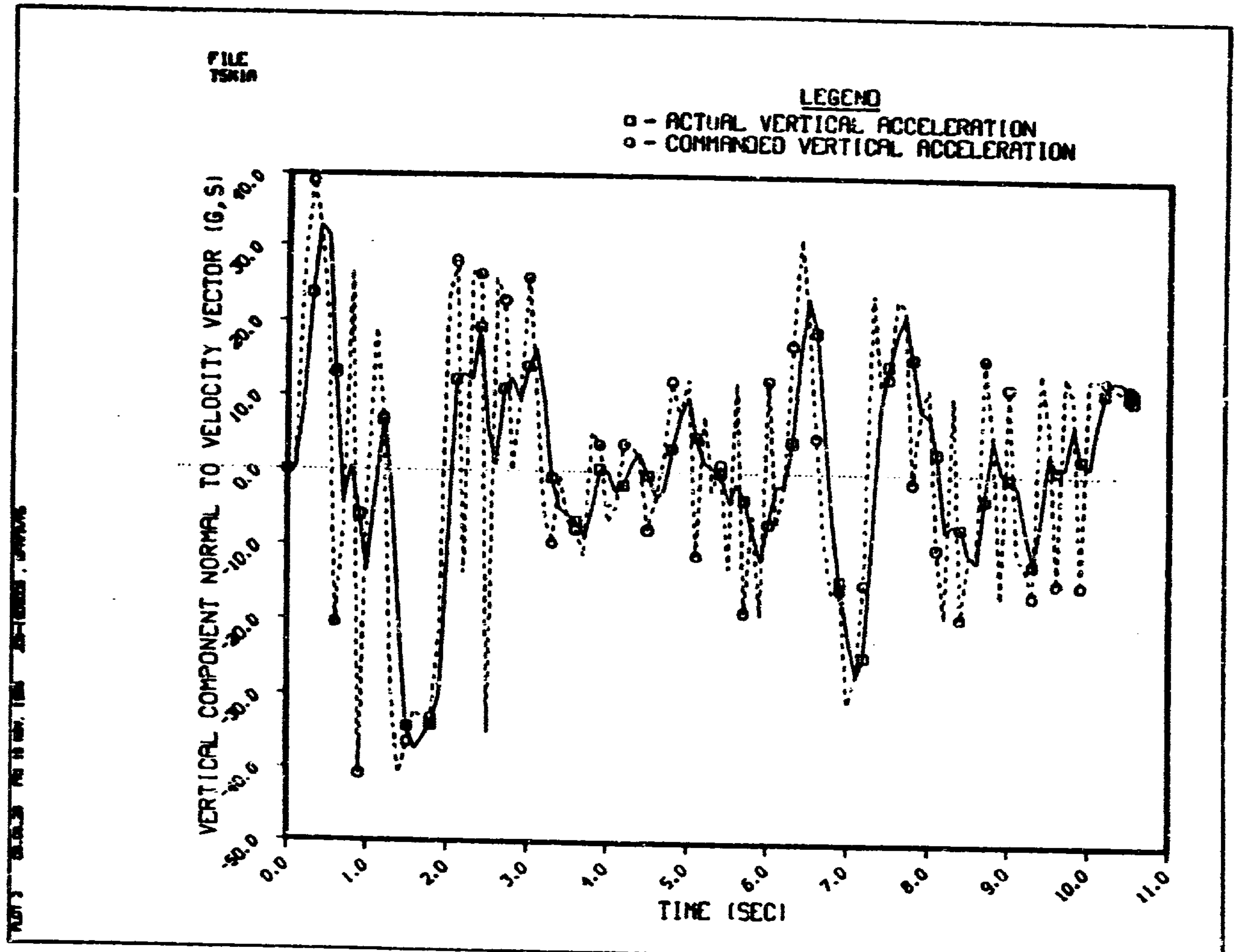


Figure E-15. Acomd, Tail Attack, Str/Lvl Tgt, Deterministic, DT = .1 sec

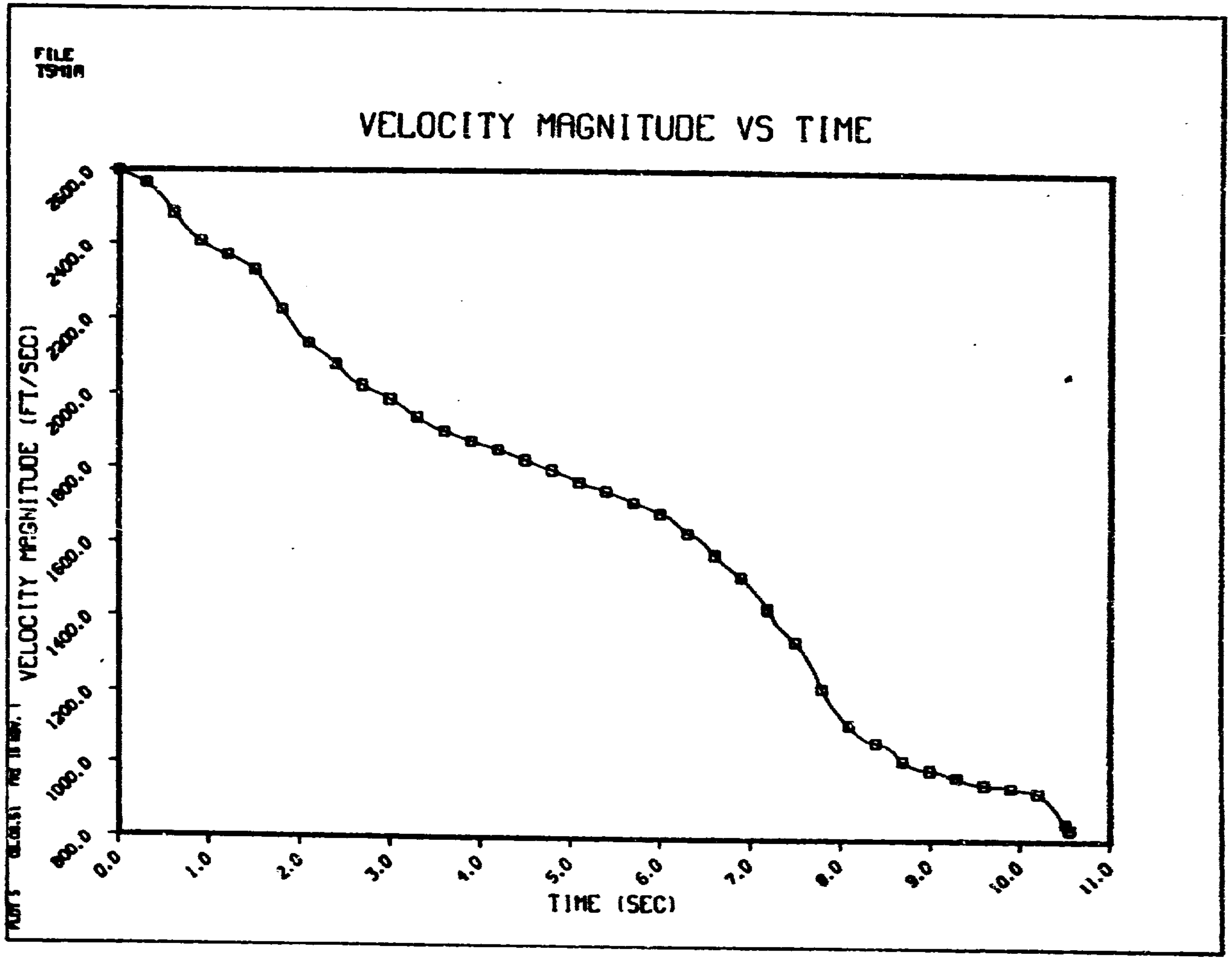


Figure E-16. Velocity vs Time, Tail Attack, Str/Lvl Tgt, Deterministic, DT = .1 sec

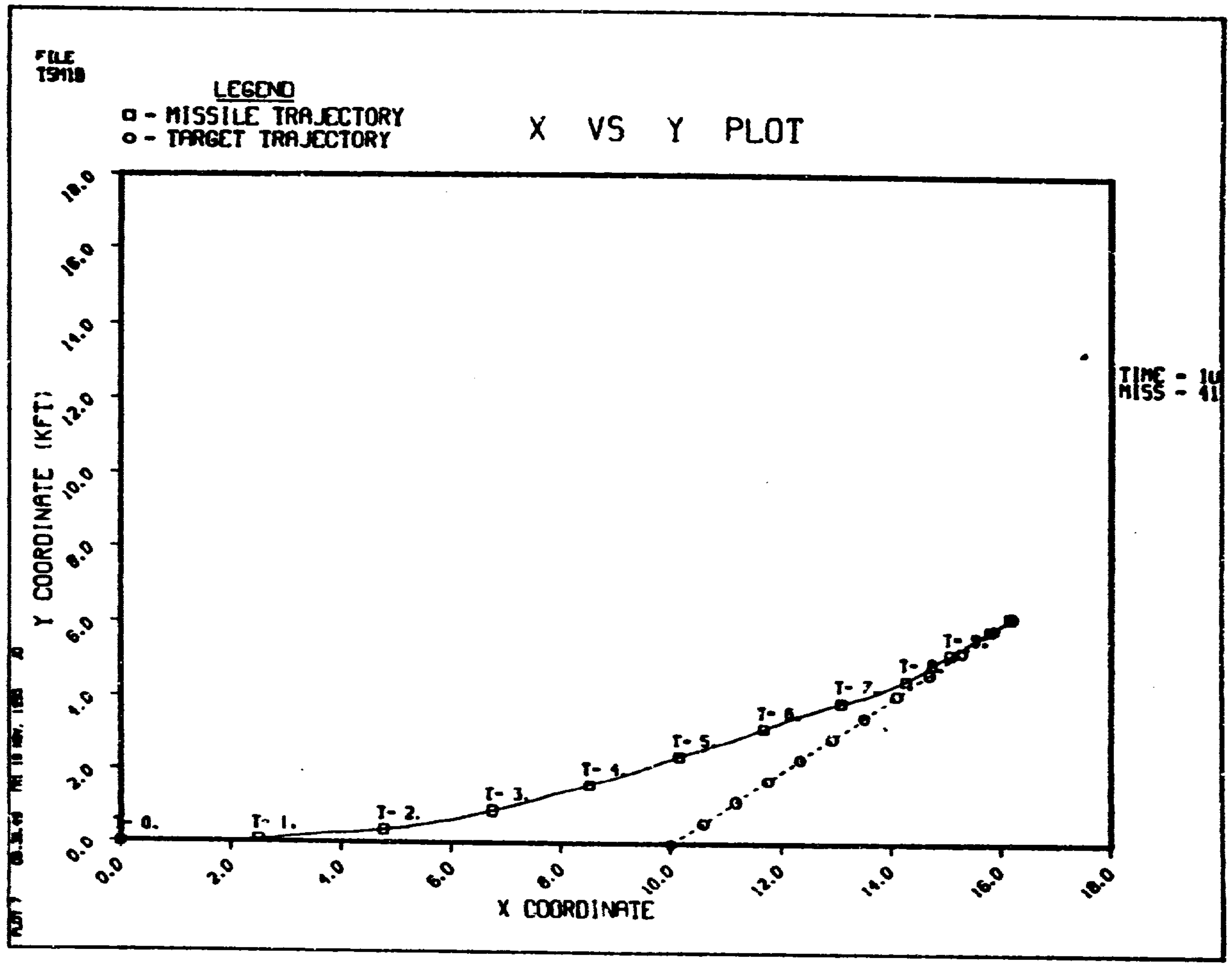


Figure E-17. X Y, Tail Attack, Turning Tgt, Deterministic, DT = .1 sec

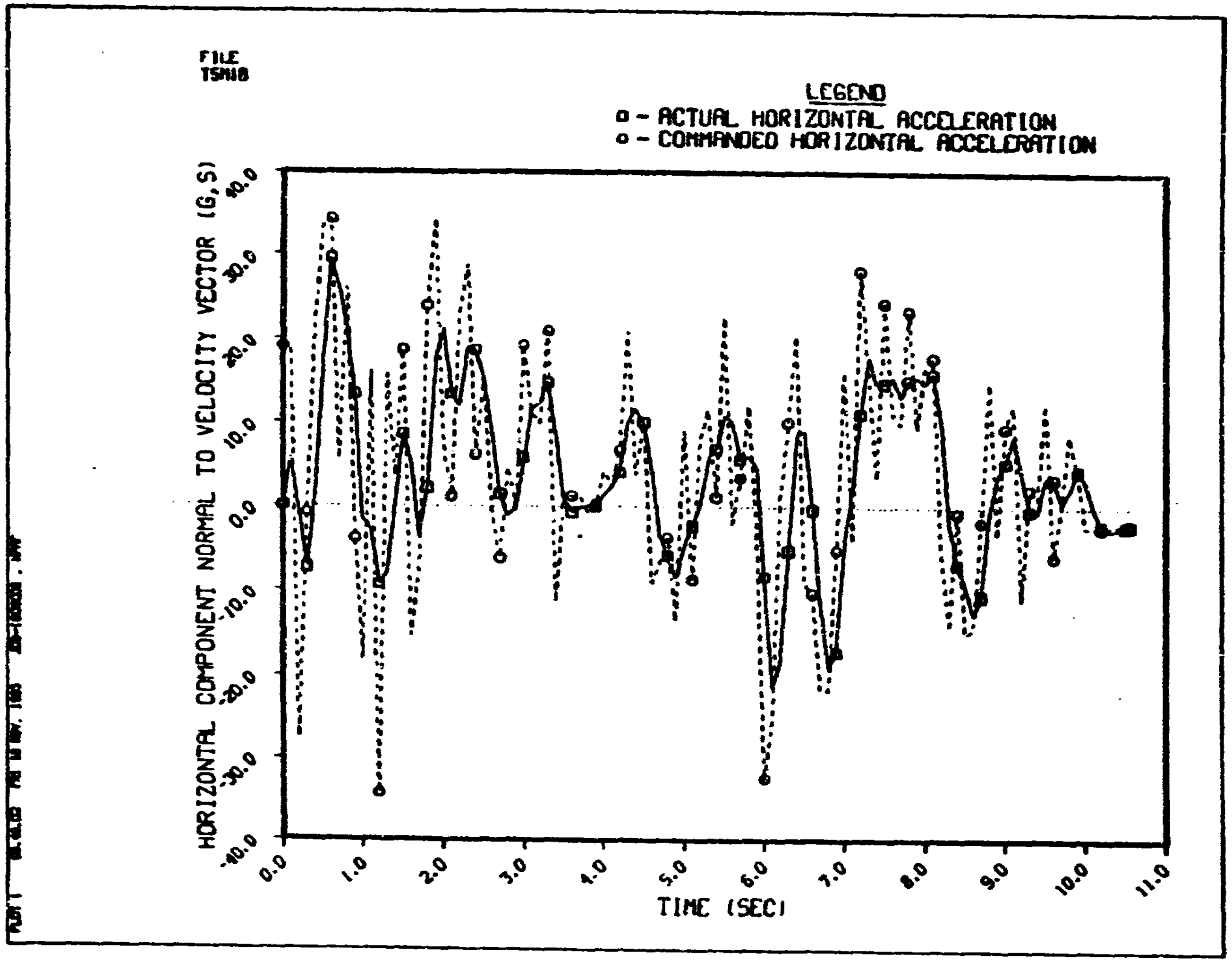


Figure E-18. Acoma, Tail Attack, Turning Tgt, Deterministic, DT = .1 sec

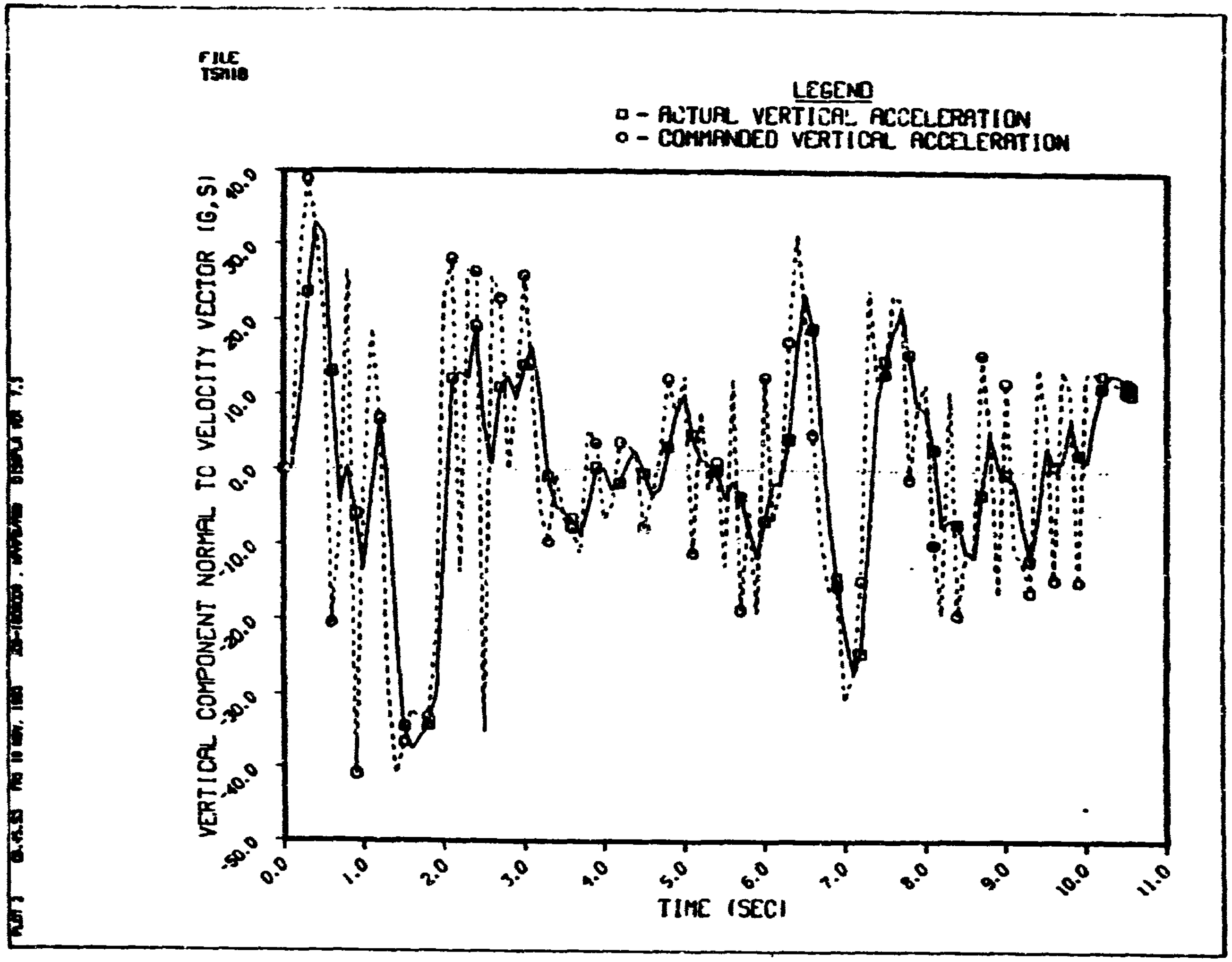


Figure E-19. Acomd, Tail Attack, Turning Tgt, Deterministic, DT = .1 sec

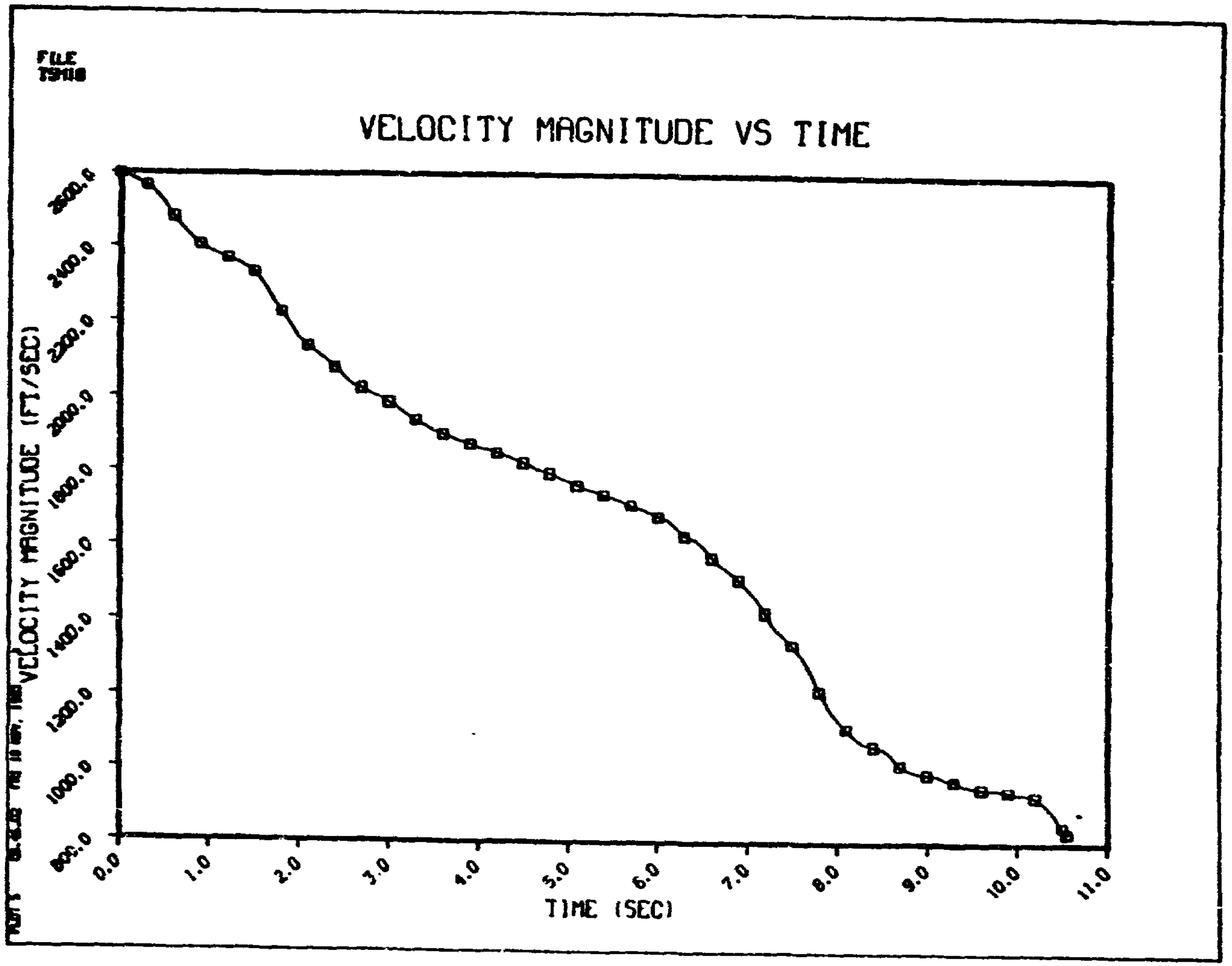


Figure E-20. Velocity vs Time, Tail Attack, Turning Tgt, Deterministic, DT = .1 sec

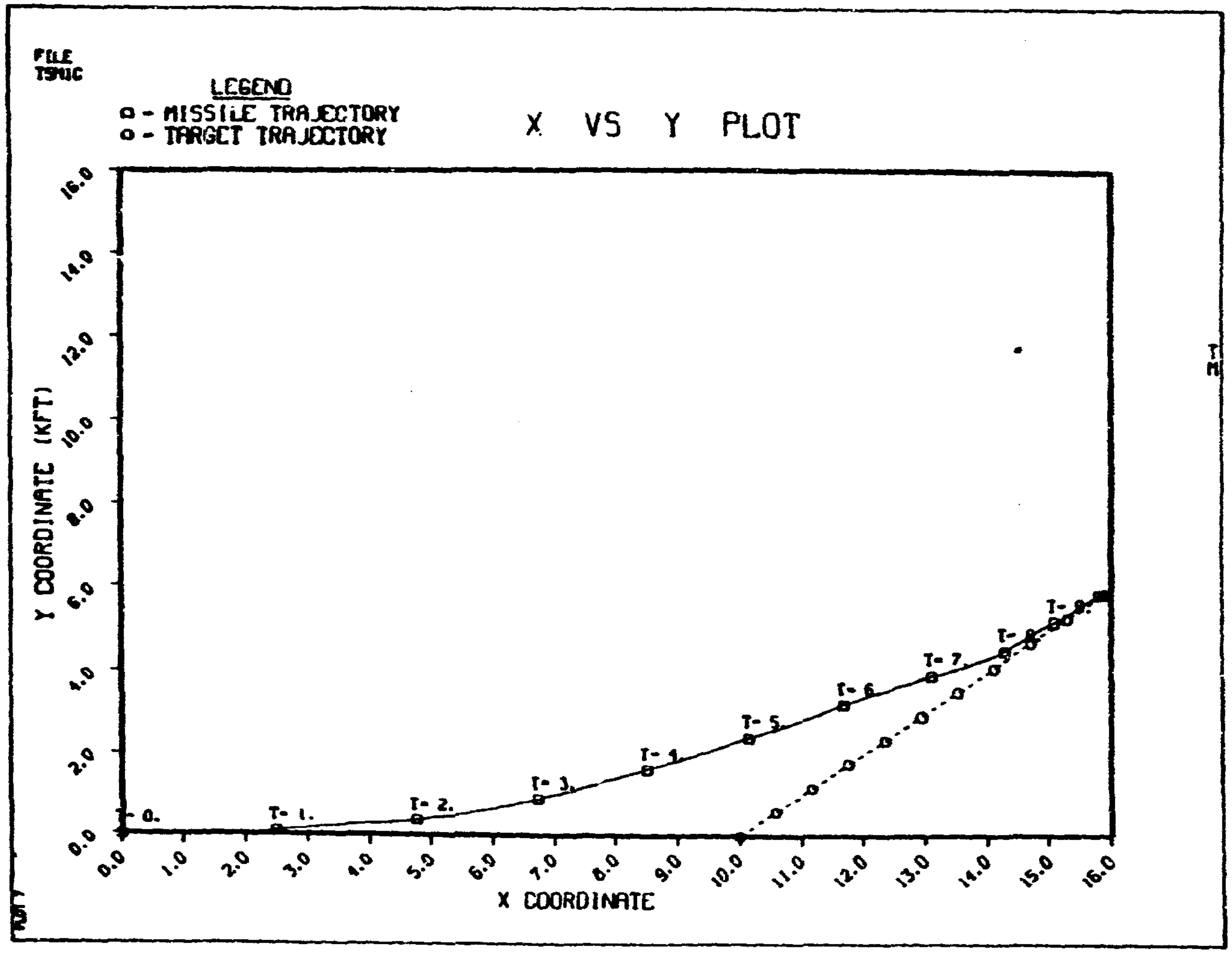


Figure E-21. X Y, Tail Attack, Climb/Dive Tgt, Deterministic, DT = .1 sec

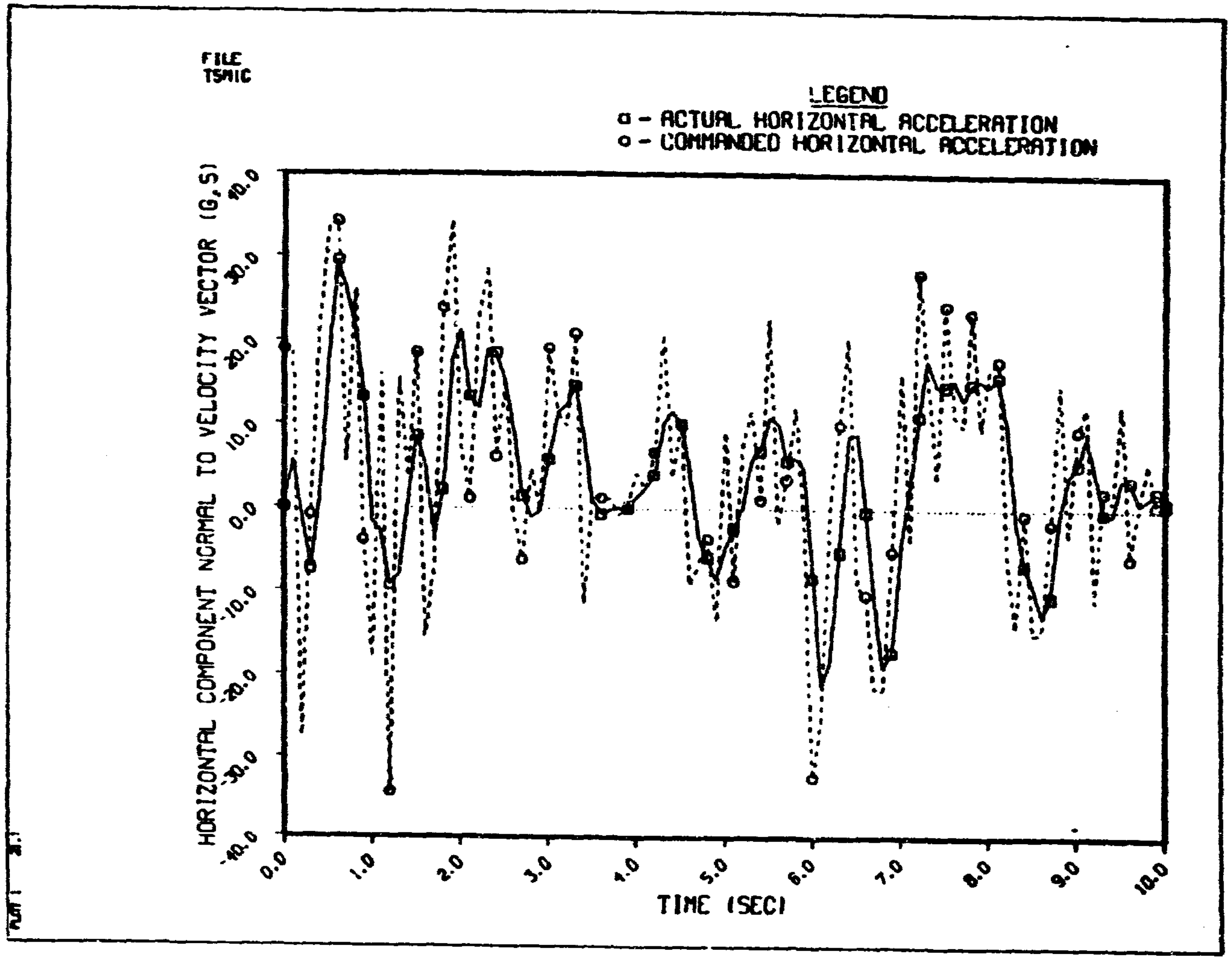


Figure E-22. Acoma, Tail Attack, Climb/Dive Tgt, Deterministic, DT = .1 sec

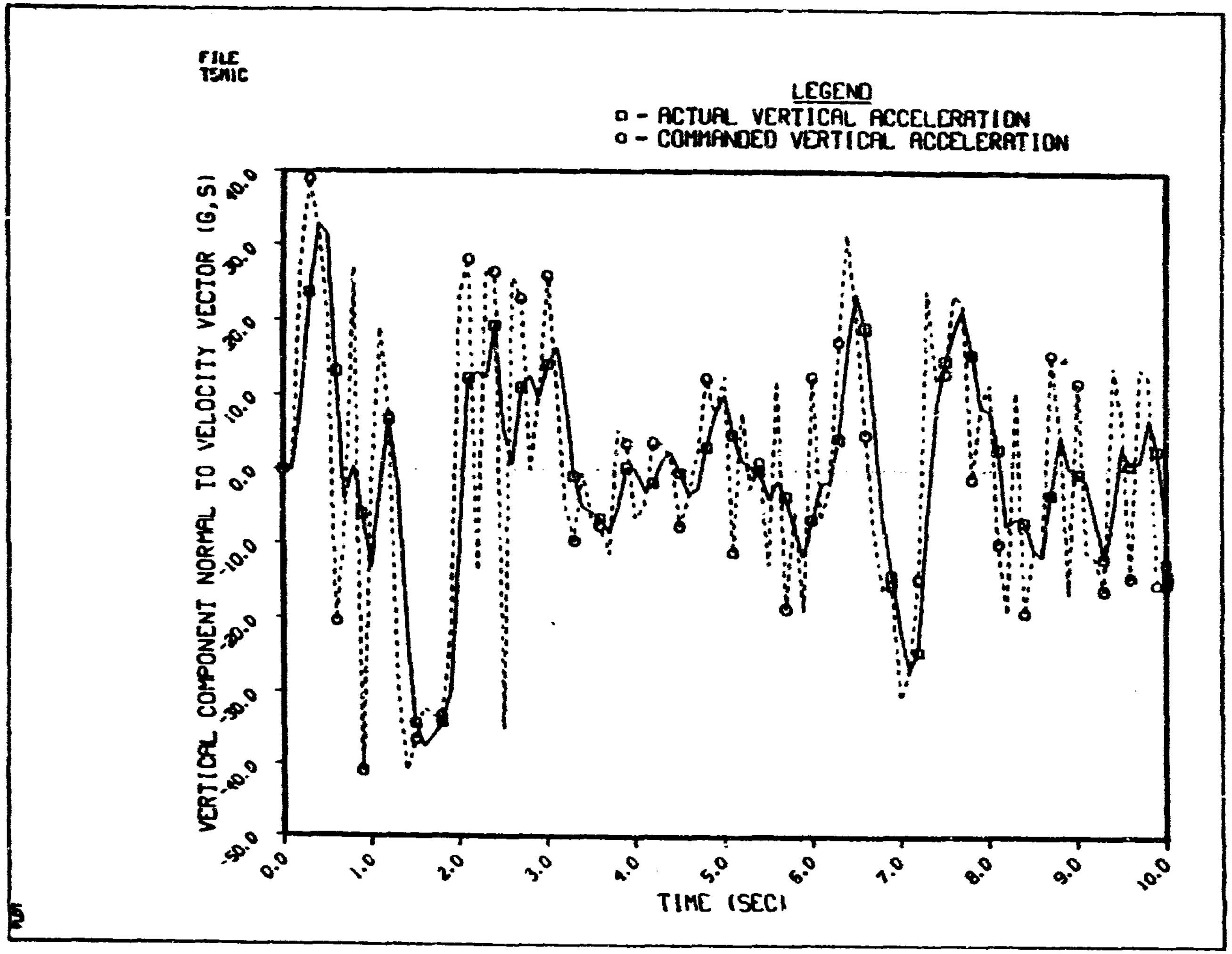


Figure E-23. Acomd, Tail Attack, Climb/Dive Tgt, Deterministic, DT = .1 sec

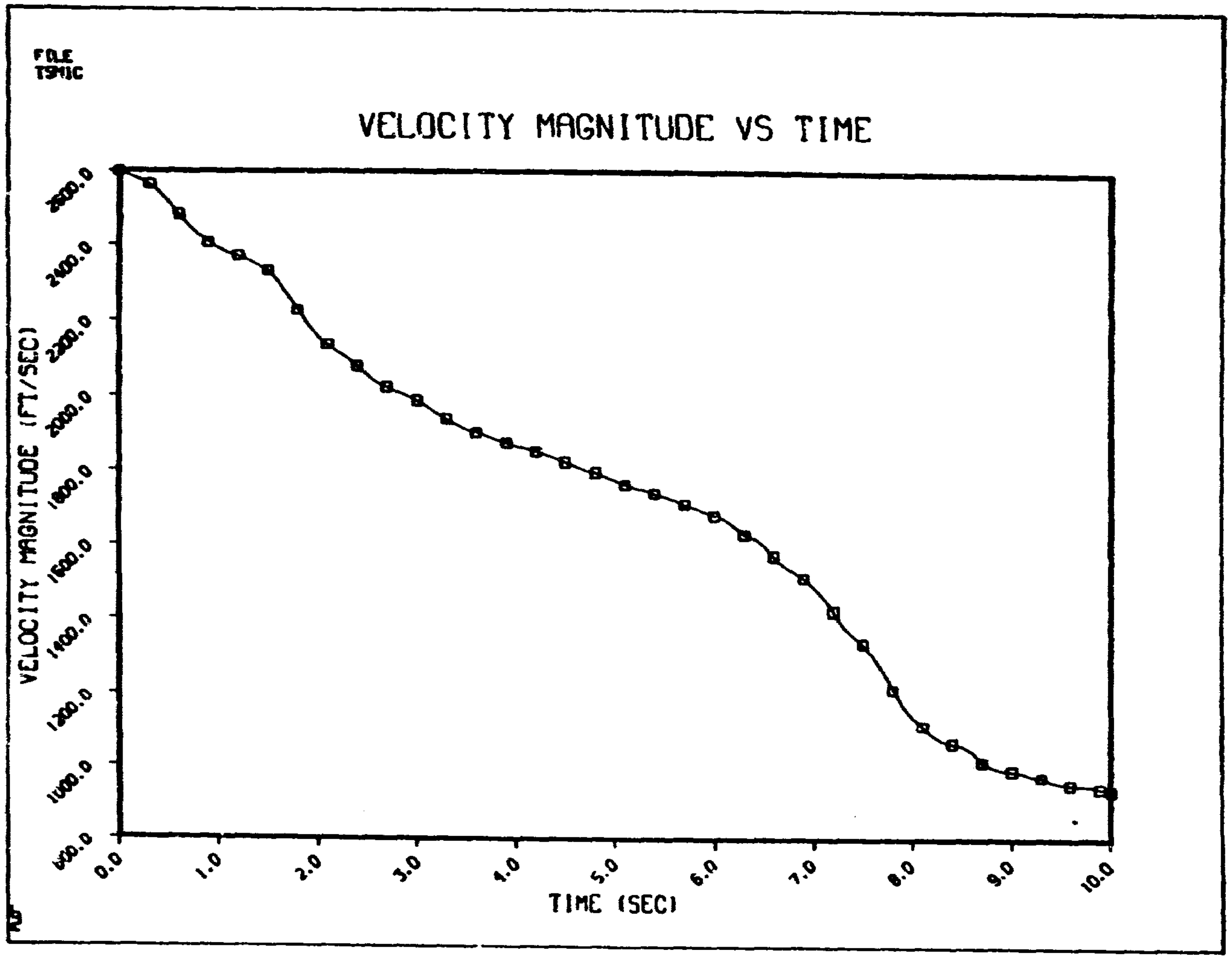


Figure E-24. Velocity vs Time, Tail Attack, Climb/Dive Tgt, Deterministic, DT = .1 sec

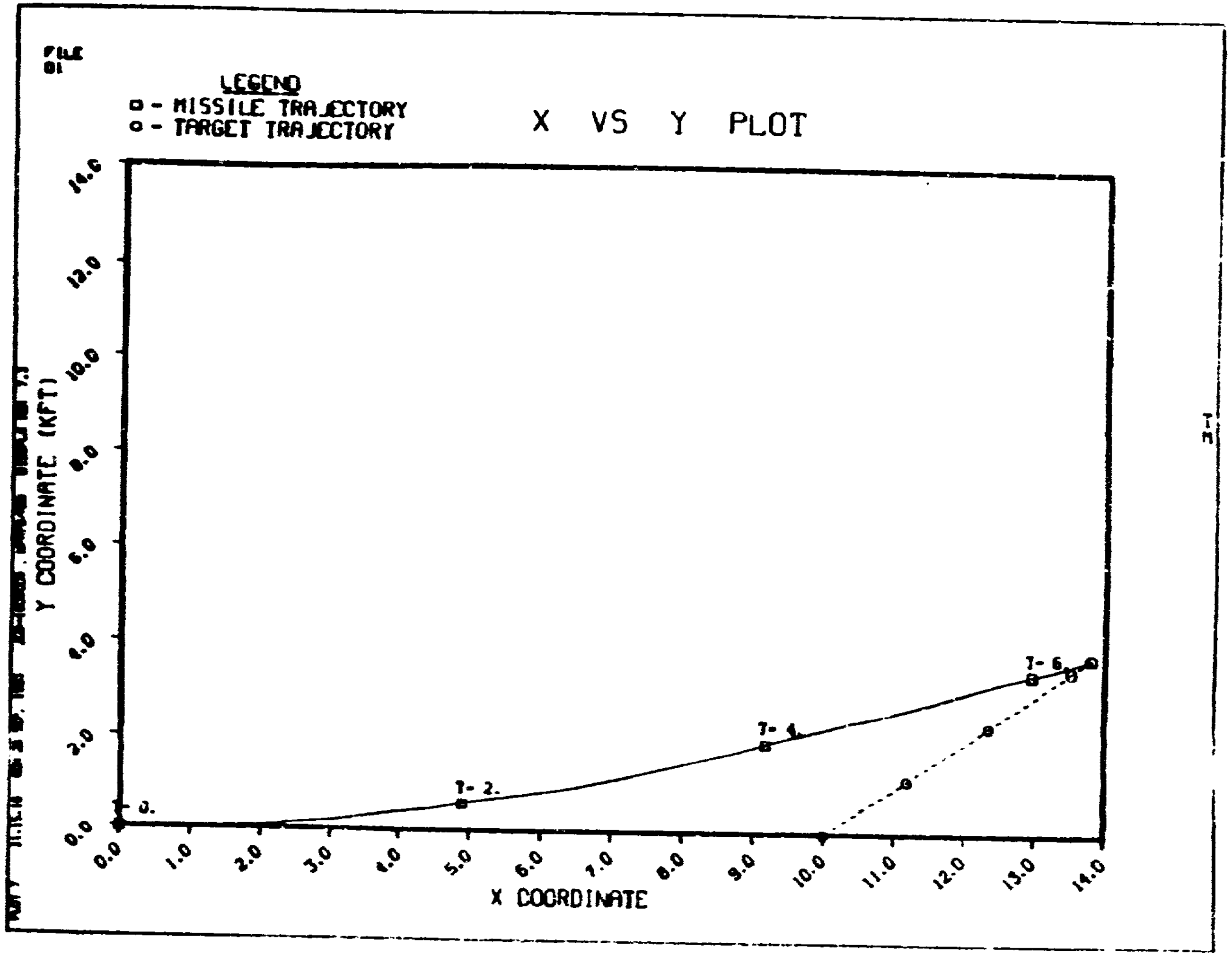


Figure E-25. X Y, Tail Attack, Str/Lvl Tgt, Stochastic, DT = .01 sec

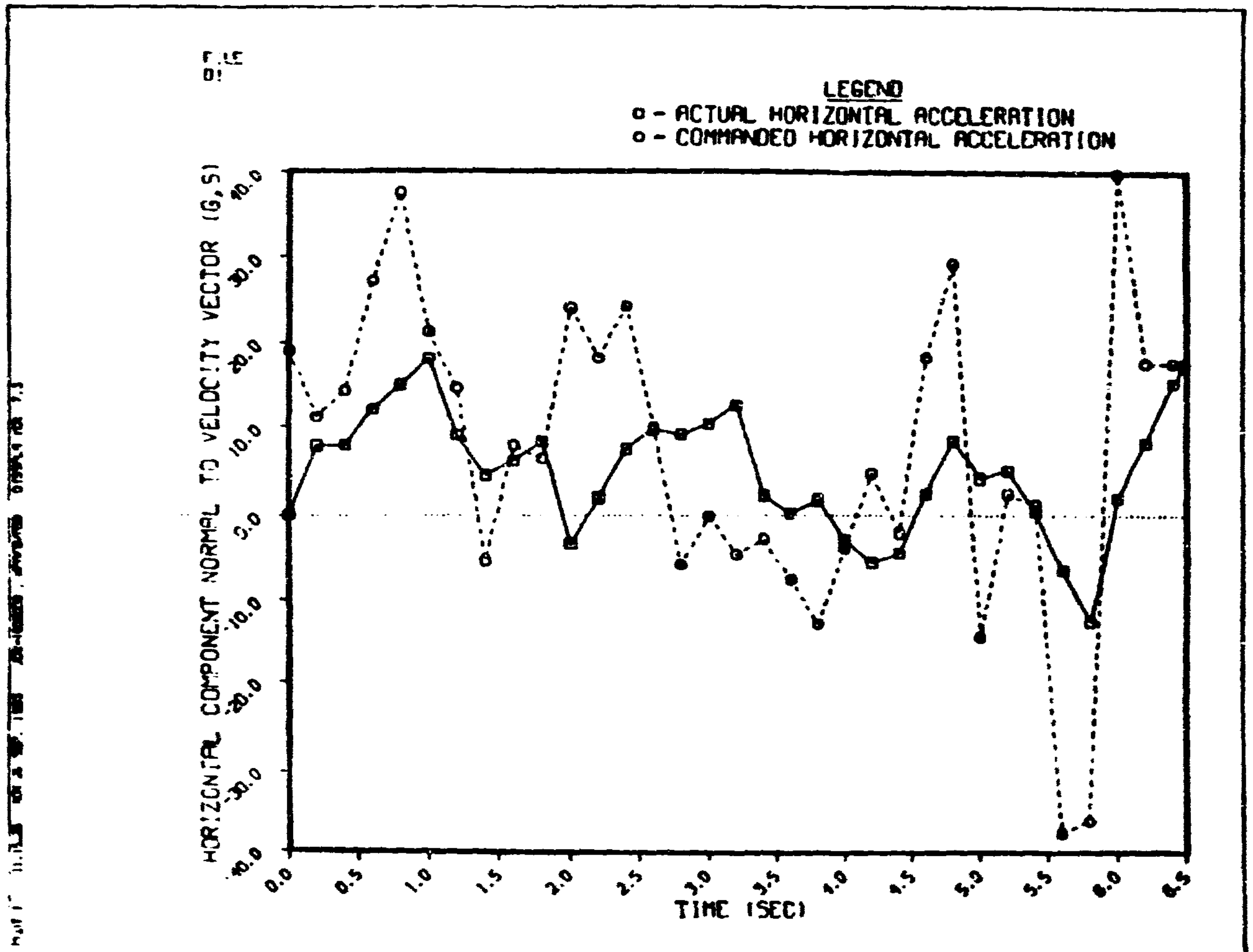


Figure E-26. Acoma, Tail Attack, Str/Lvl Tgt, Stochastic, DT = .01 sec

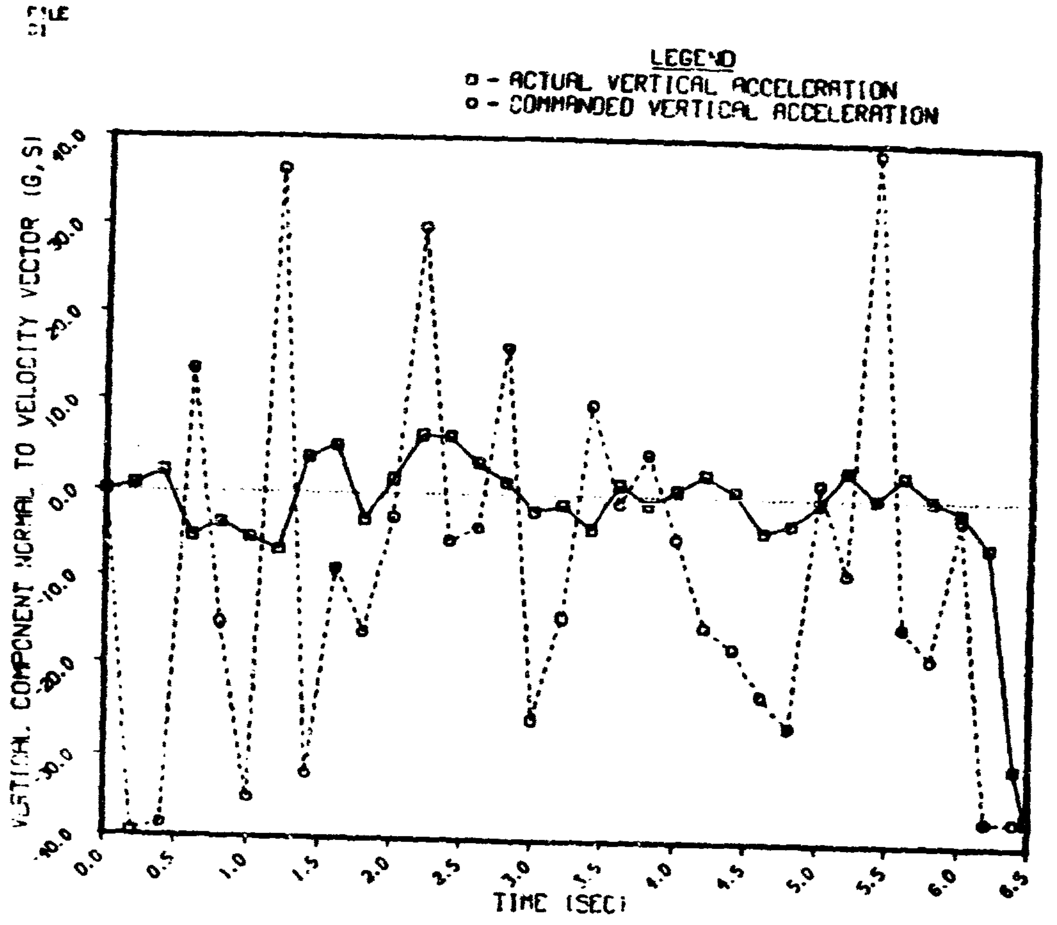


Figure E-27. Acomd, Tail Attack, Str/Lvl Tgt, Stochastic, DT = .01 sec

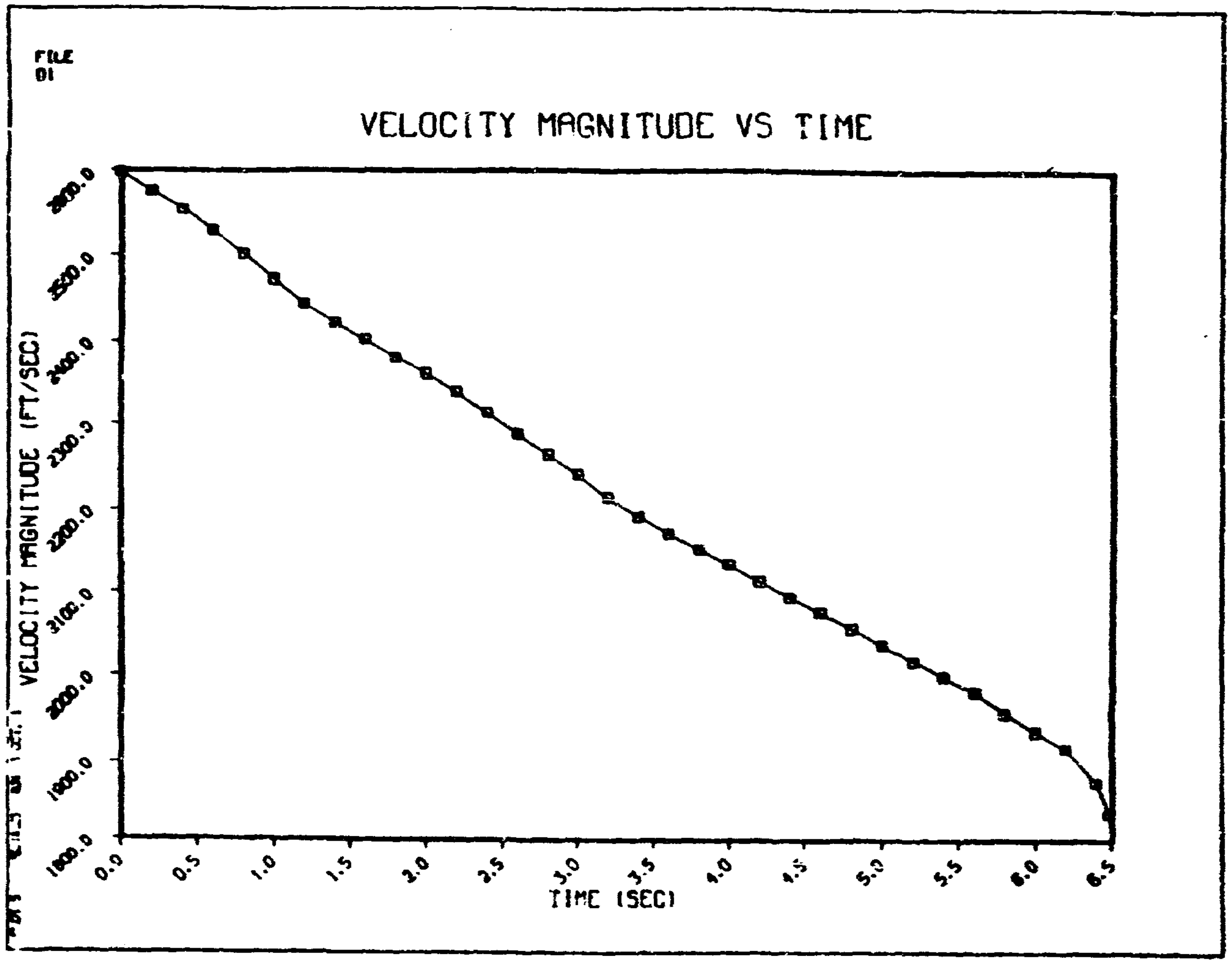


Figure E-28. Velocity vs Time, Tail Attack, Str/Lvl Tgt, Stochastic, DT = .01 sec

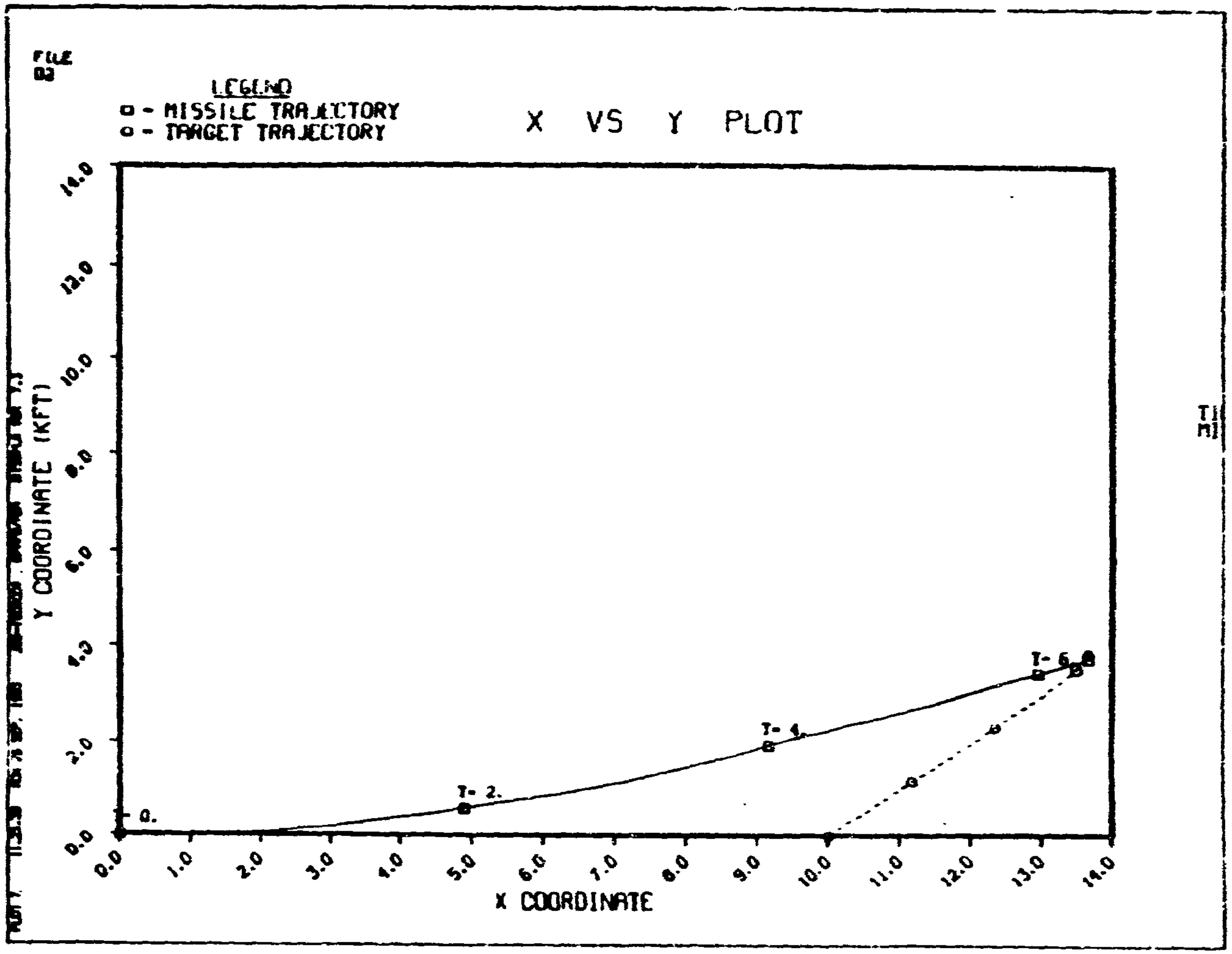


Figure E-29. X Y, Tail Attack, Turning Tgt, Stochastic, DT = .01 sec

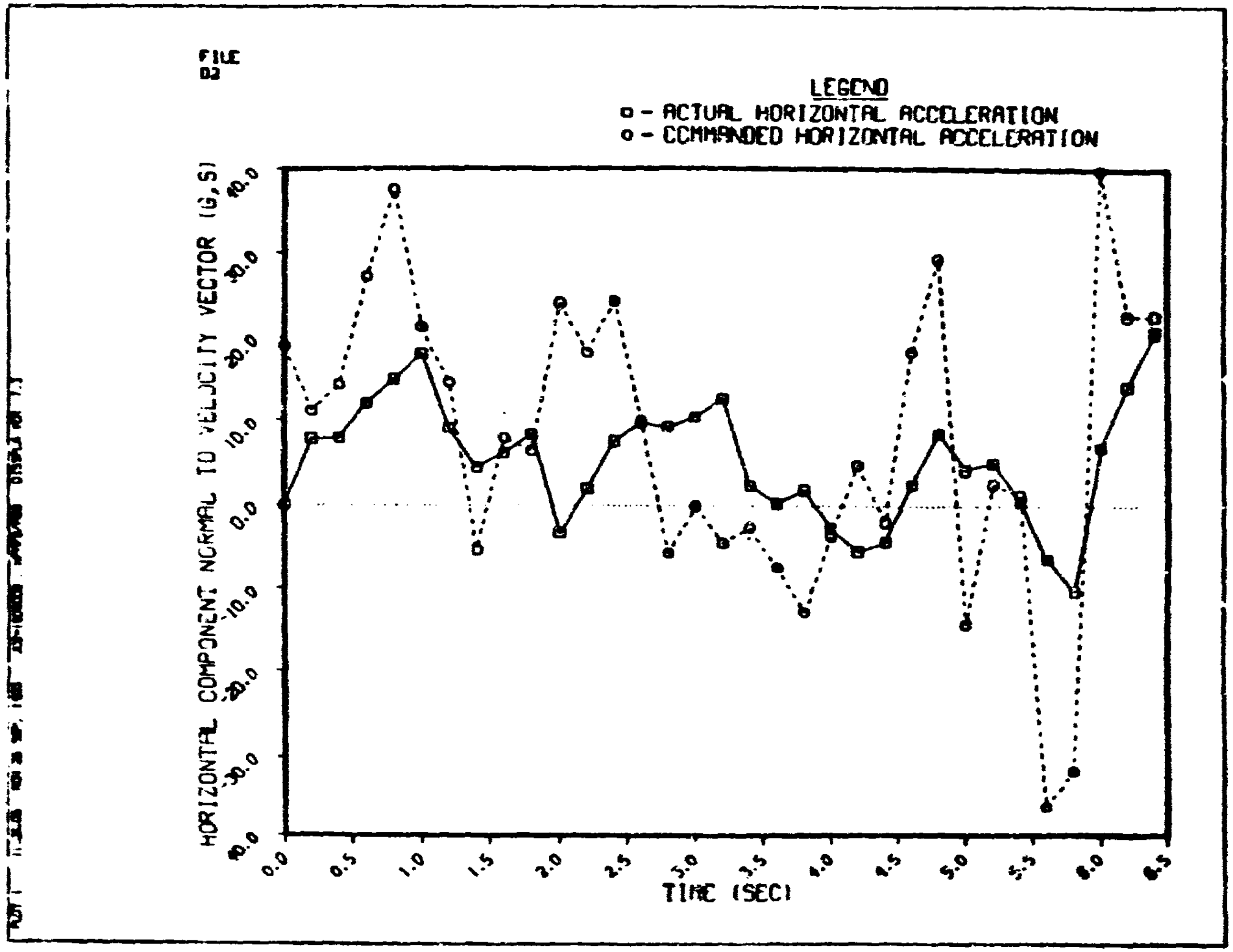


Figure E-30. Acoma, Tail Attack, Turning Tgt, Stochastic, DT = .01 sec

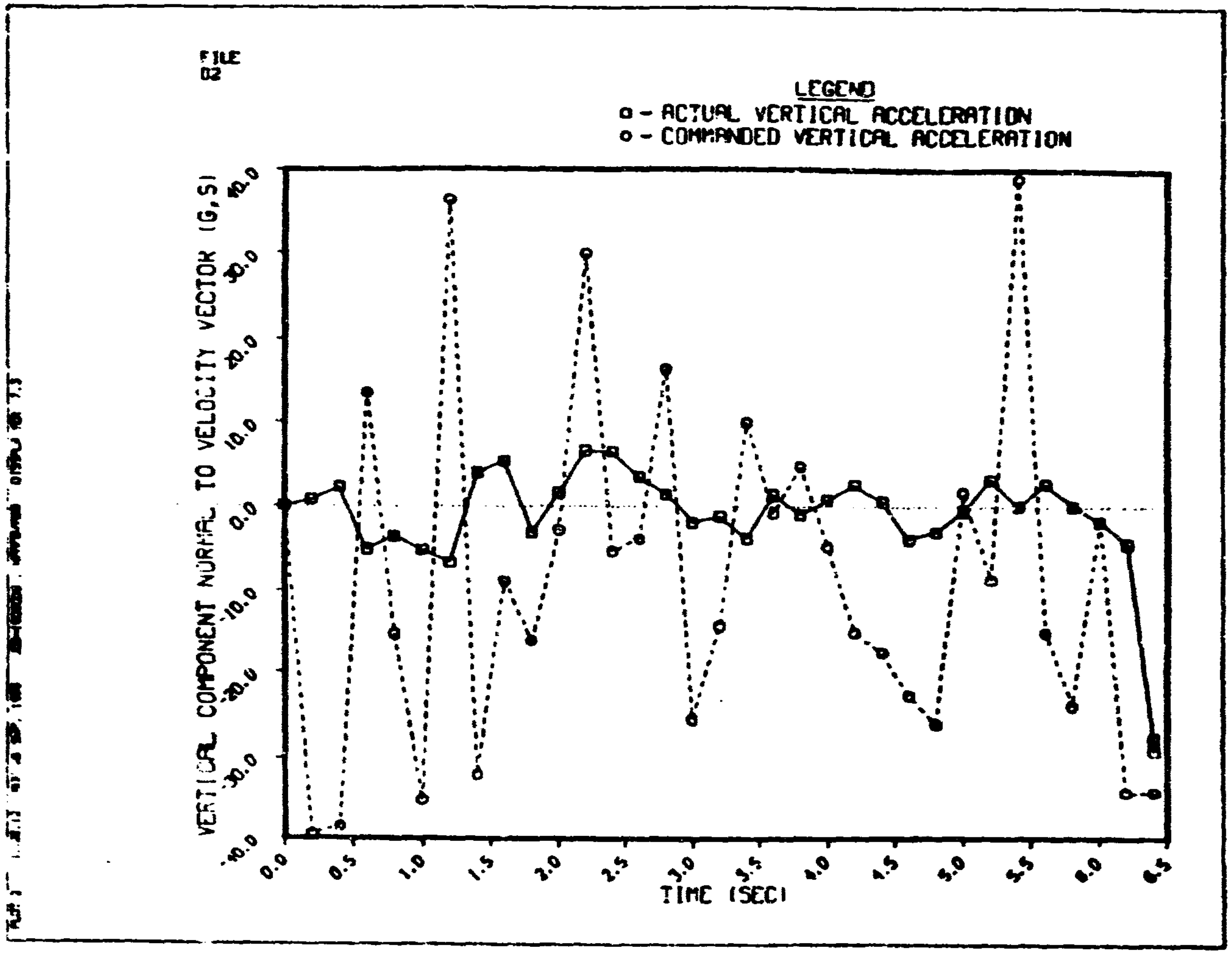


Figure E-31. Acomd, Tail Attack, Turning Tgt, Stochastic, DT = .01 sec

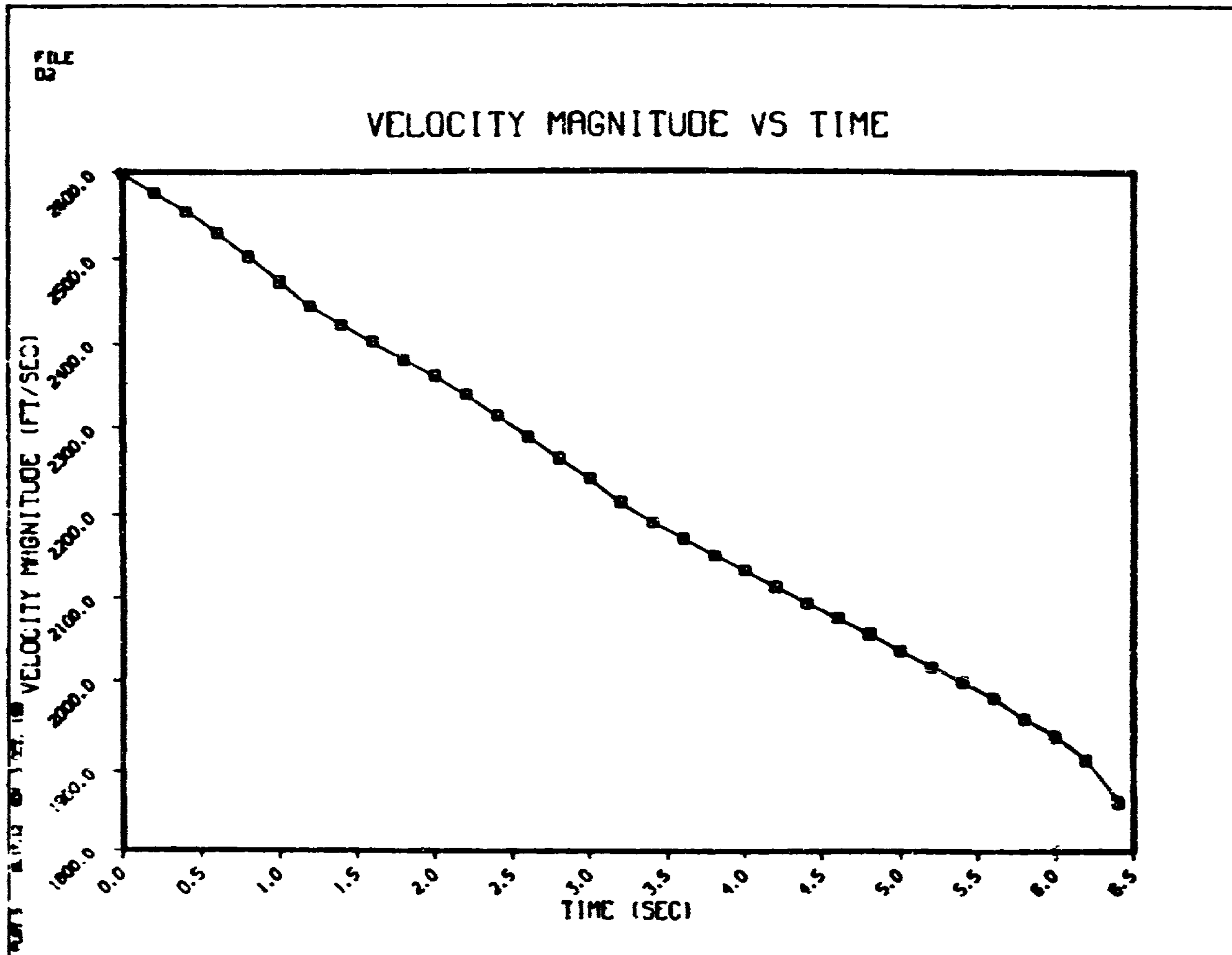


Figure E-32. Velocity vs Time, Tail Attack, Turning Tgt, Stochastic, DT = .01 sec

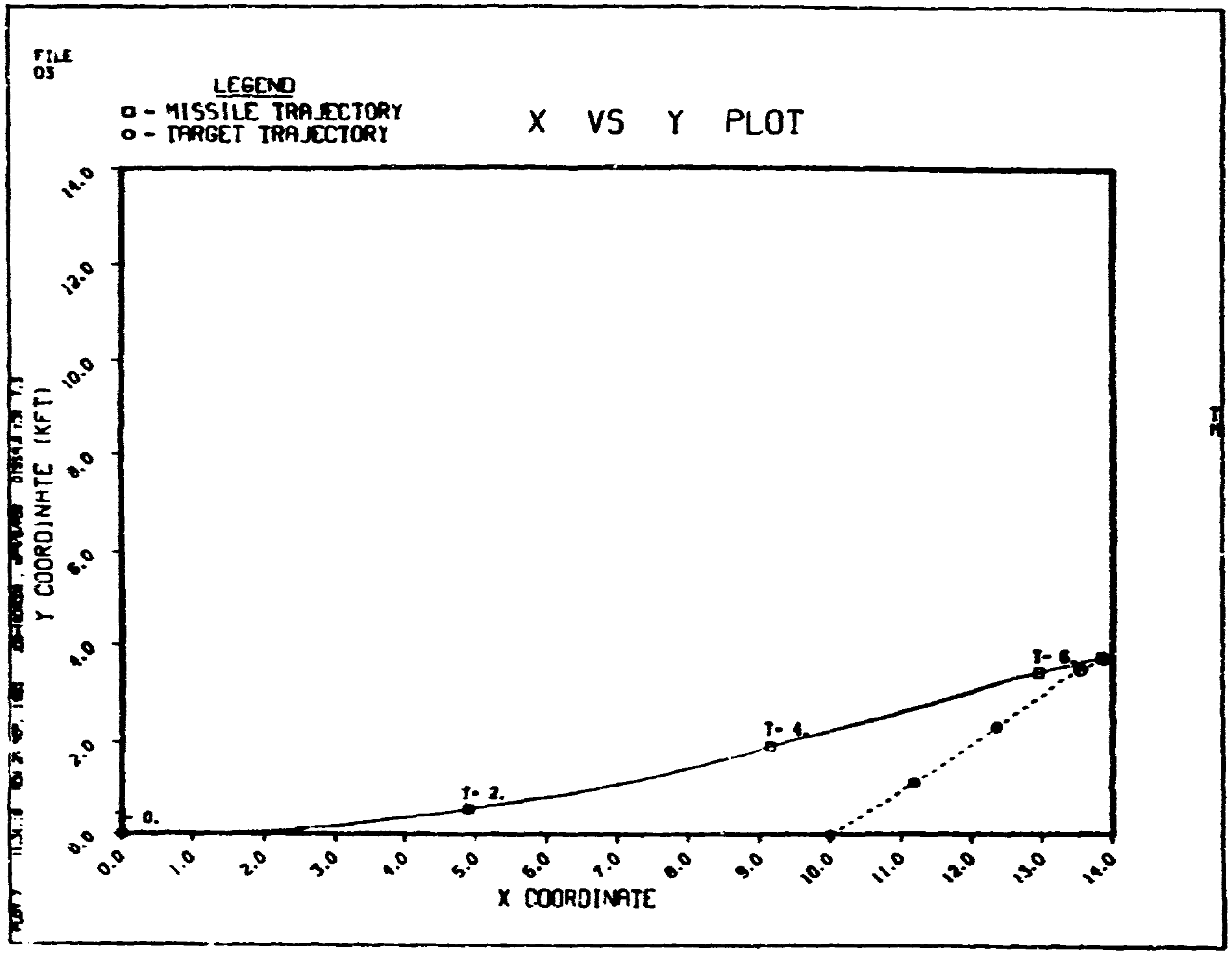


Figure E-33. X Y, Tail Attack, Climb/Dive Tgt, Stochastic, DT = .01 sec

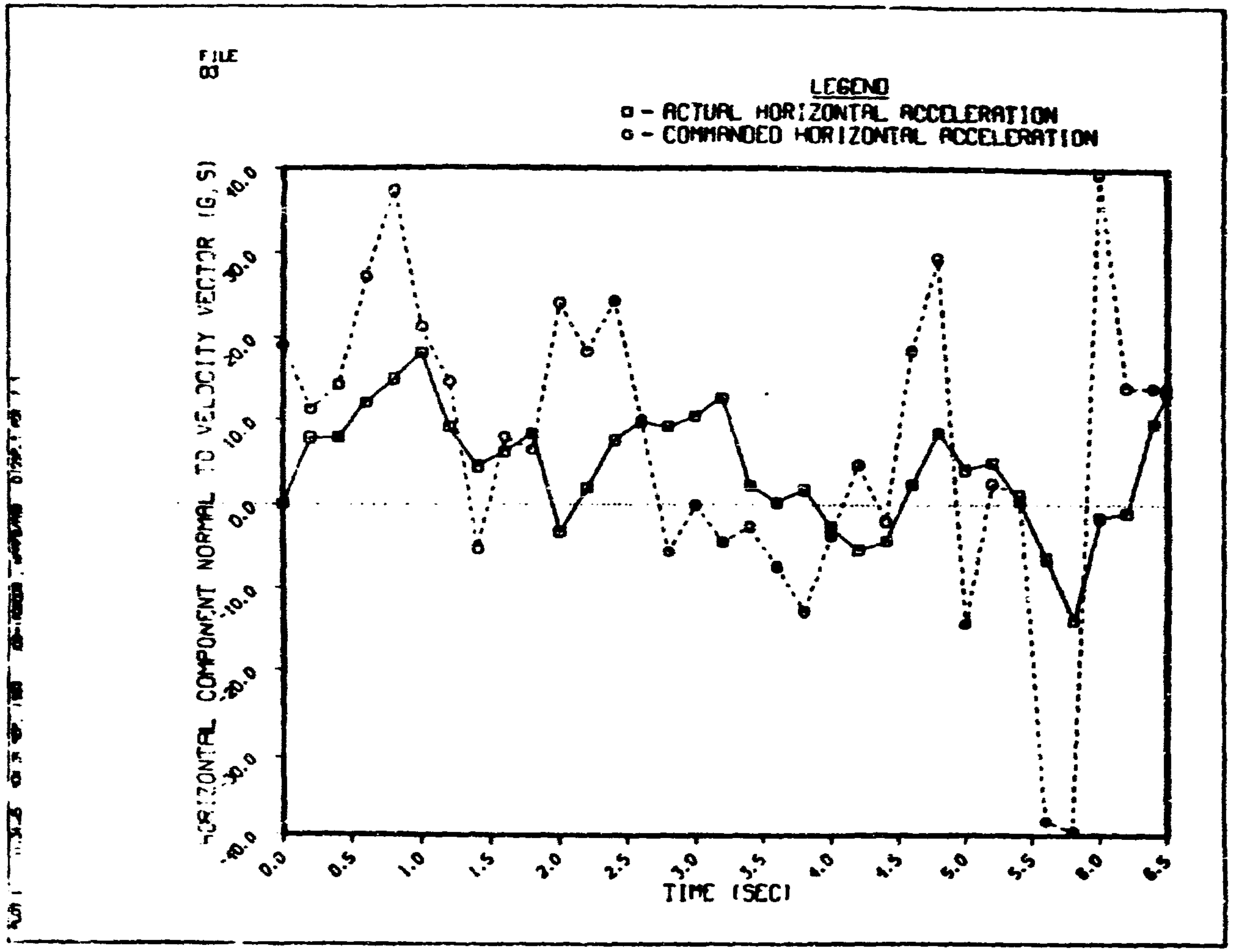


Figure E-34. Acoma, Tail Attack, Climb/Dive Tgt, Stochastic, DT = .01 sec

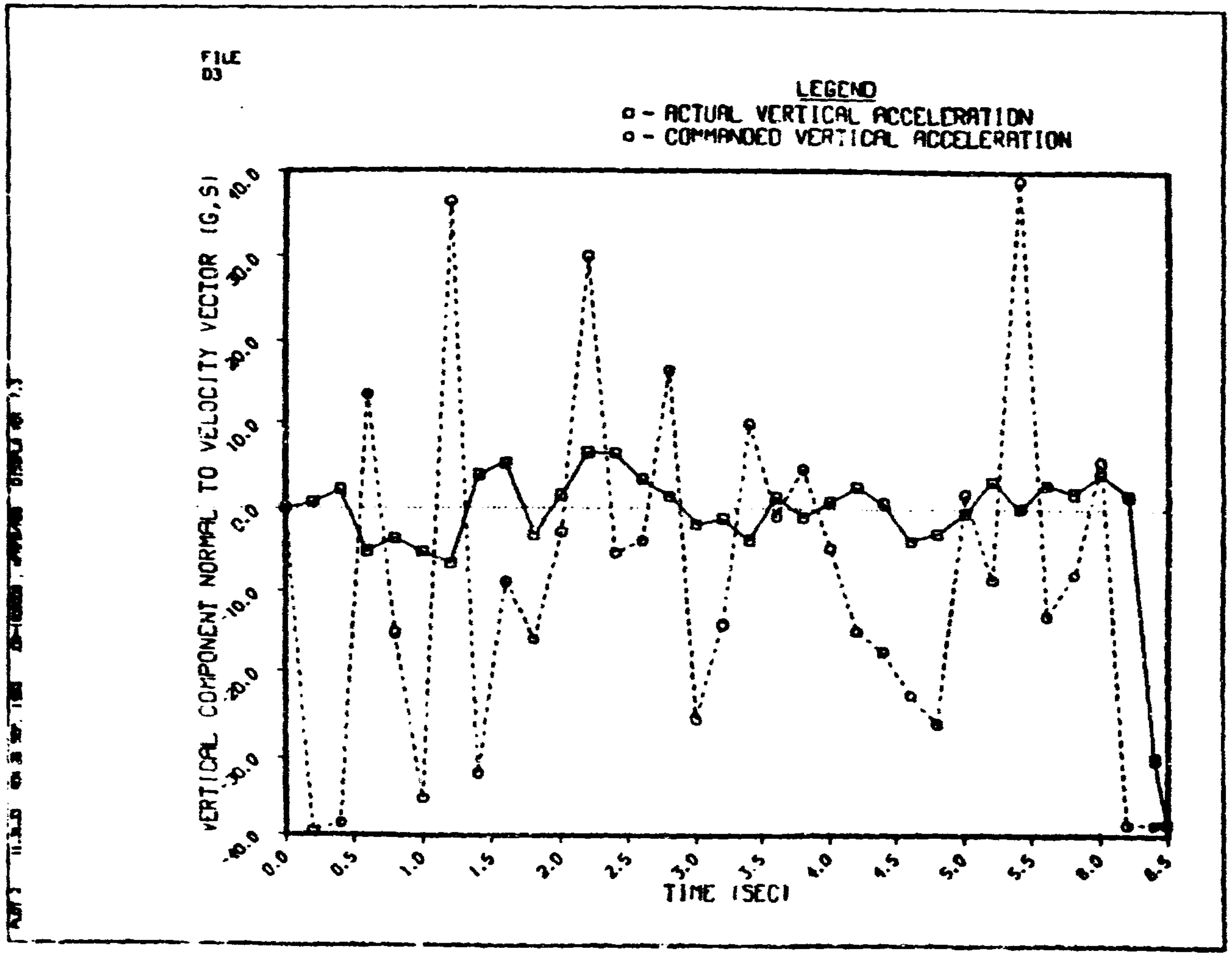


Figure E-35. Acomd, Tail Attack, Climb/Dive Tgt, Stochastic, DT = .01 sec

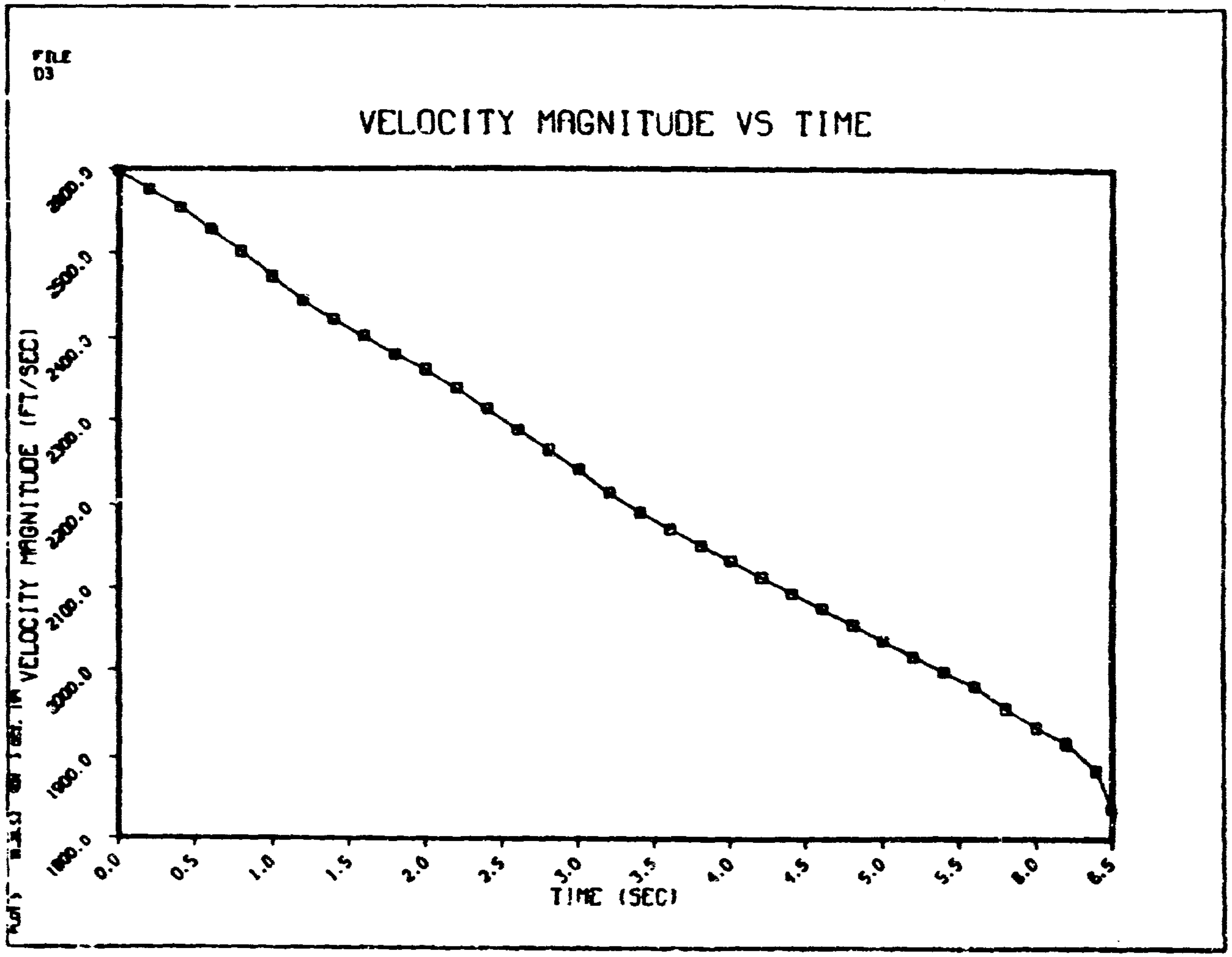


Figure E-36. Velocity vs Time, Tail Attack, Climb/Dive Tgt, Stochastic, DT = .01 sec

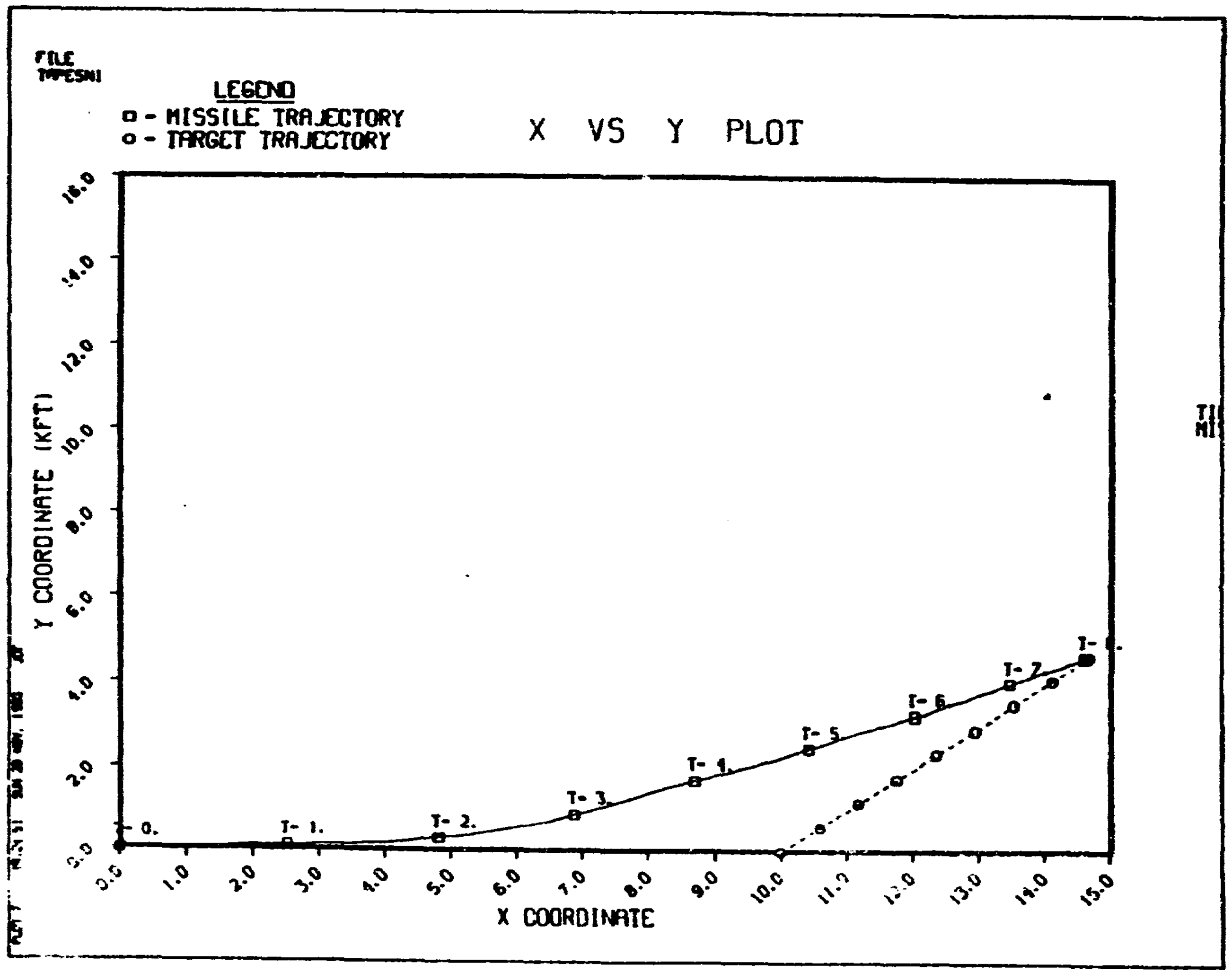


Figure E-37. X Y, Tail Attack, Str/Lvl Tgt, Stochastic, DT = .1 sec

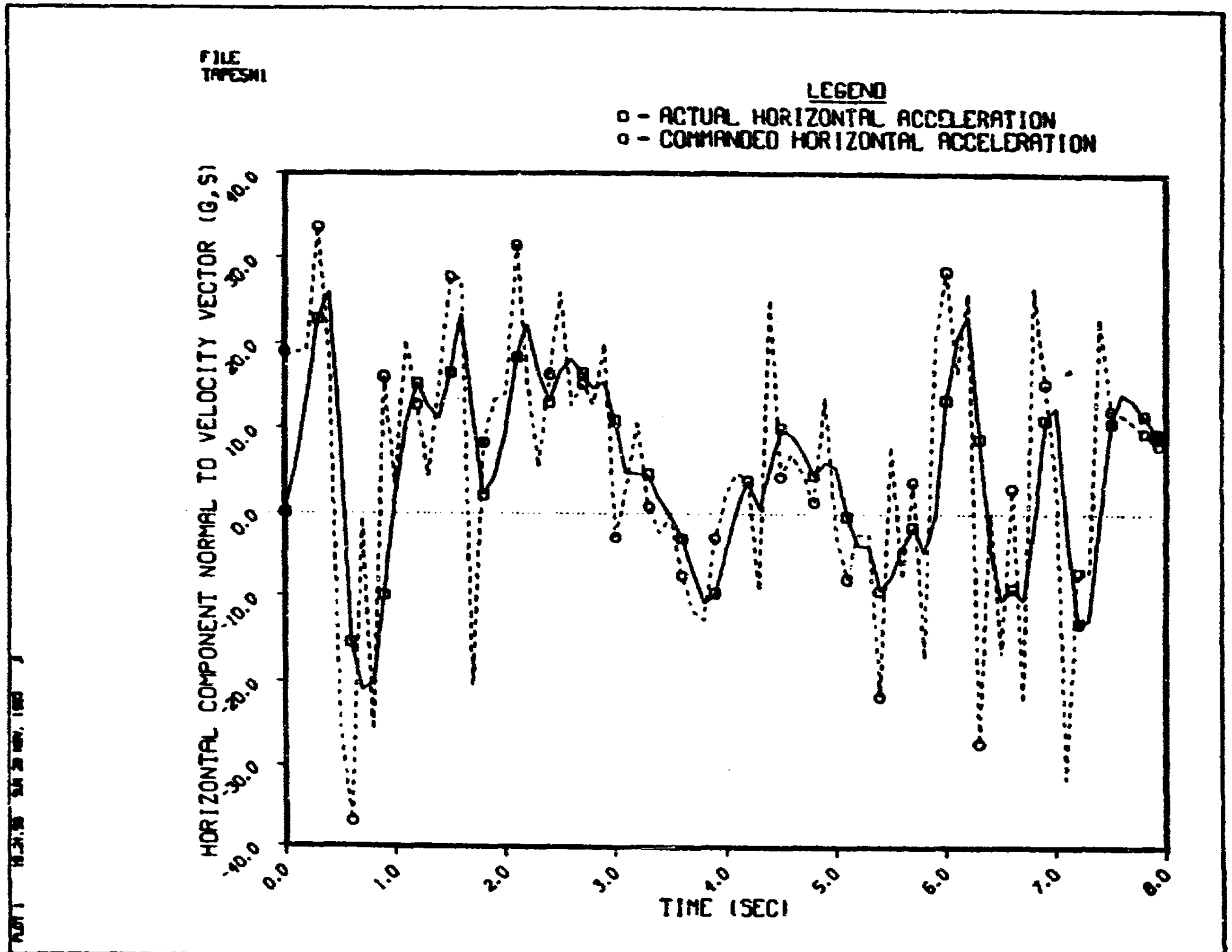


Figure E-38. Acoma, Tail Attack, Str/Lvl Tgt, Stochastic, DT = .1 sec

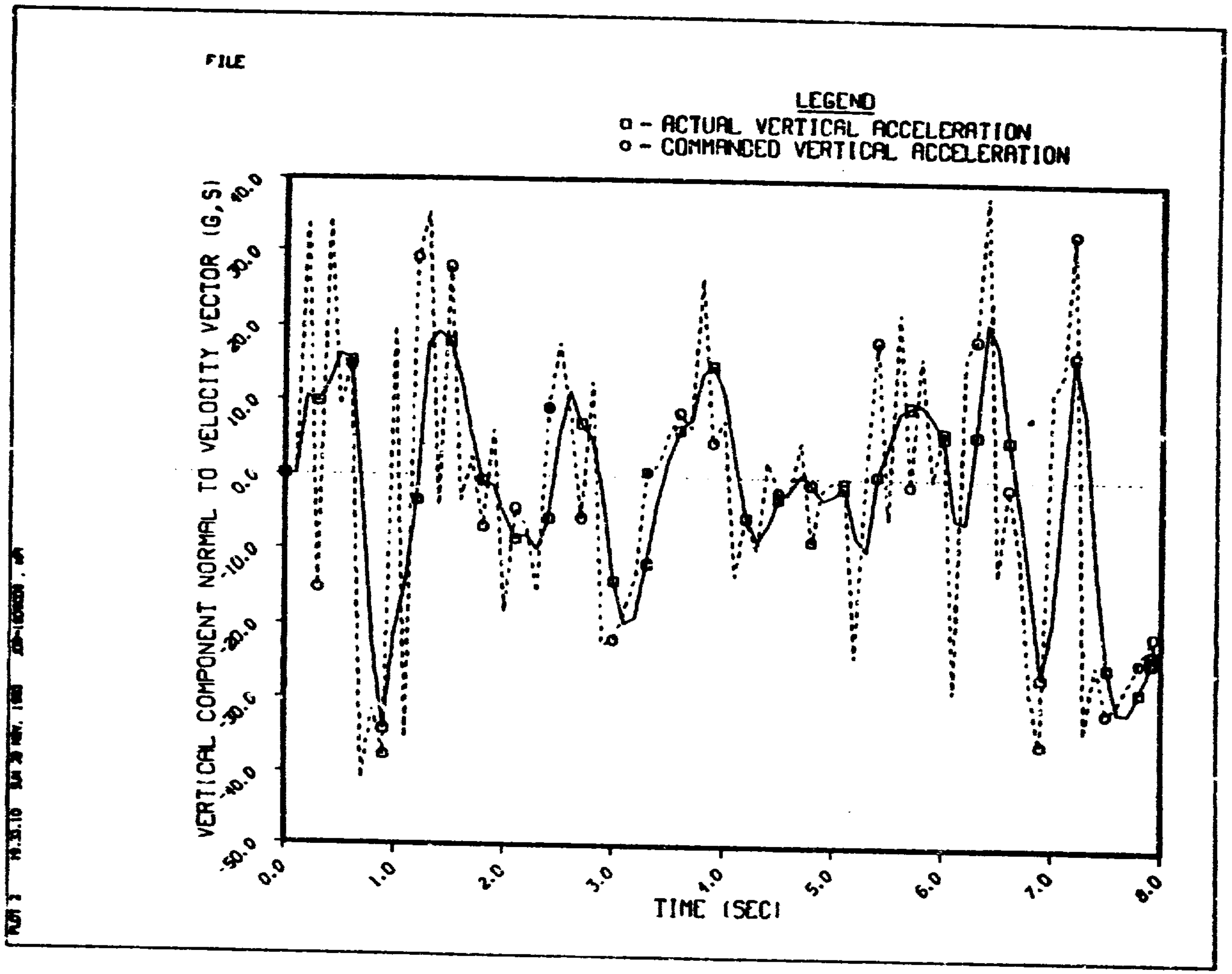


Figure E-39. Acomd, Tail Attack, Str/Lvl Tgt, Stochastic, DT = .1 sec

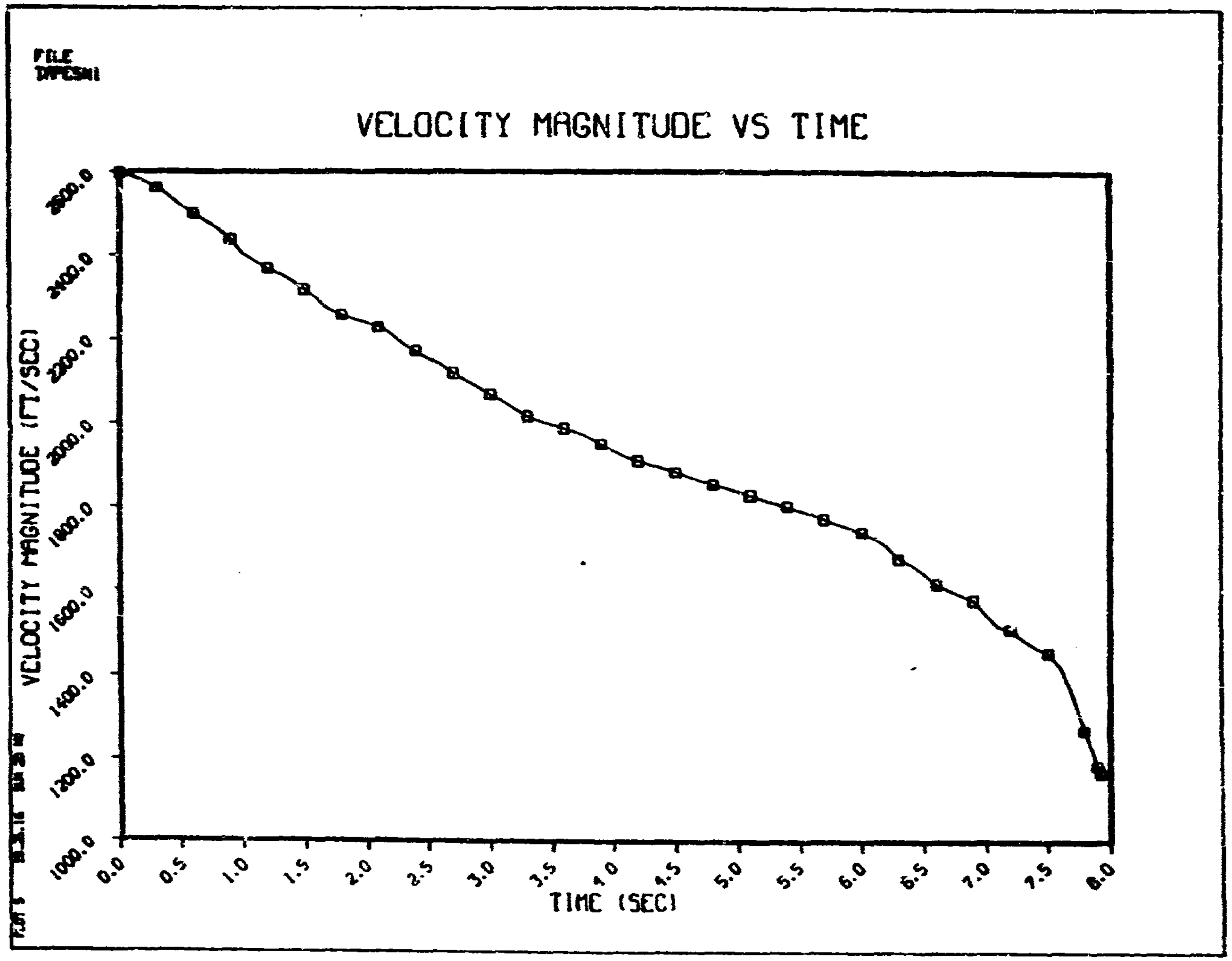


Figure E-40. Velocity vs Time, Tail Attack, Str/Lvl Tgt, Stochastic, DT = .1 sec

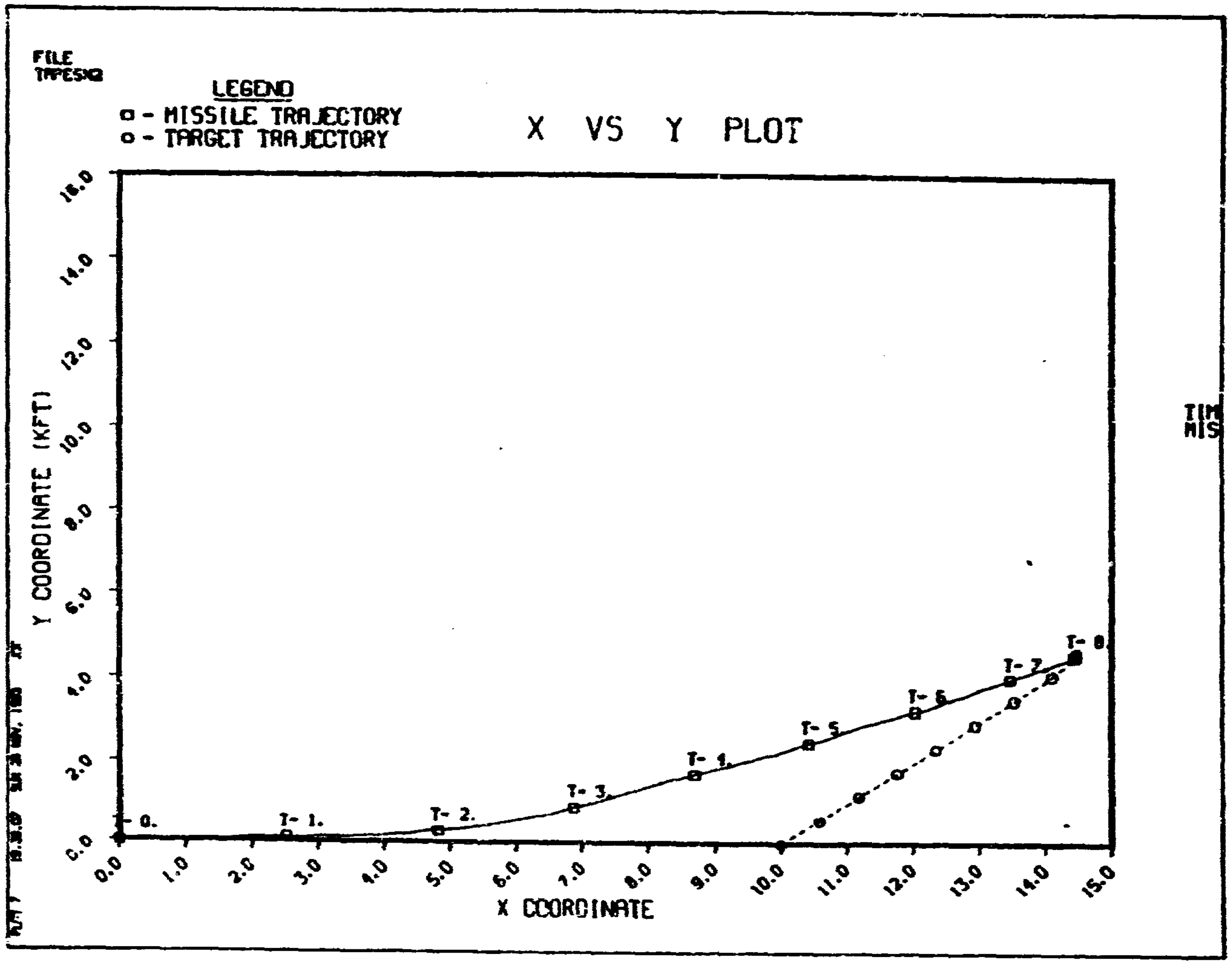


Figure E-41. X Y, Tail Attack, Turning Tgt, Stochastic, DT = .1 sec

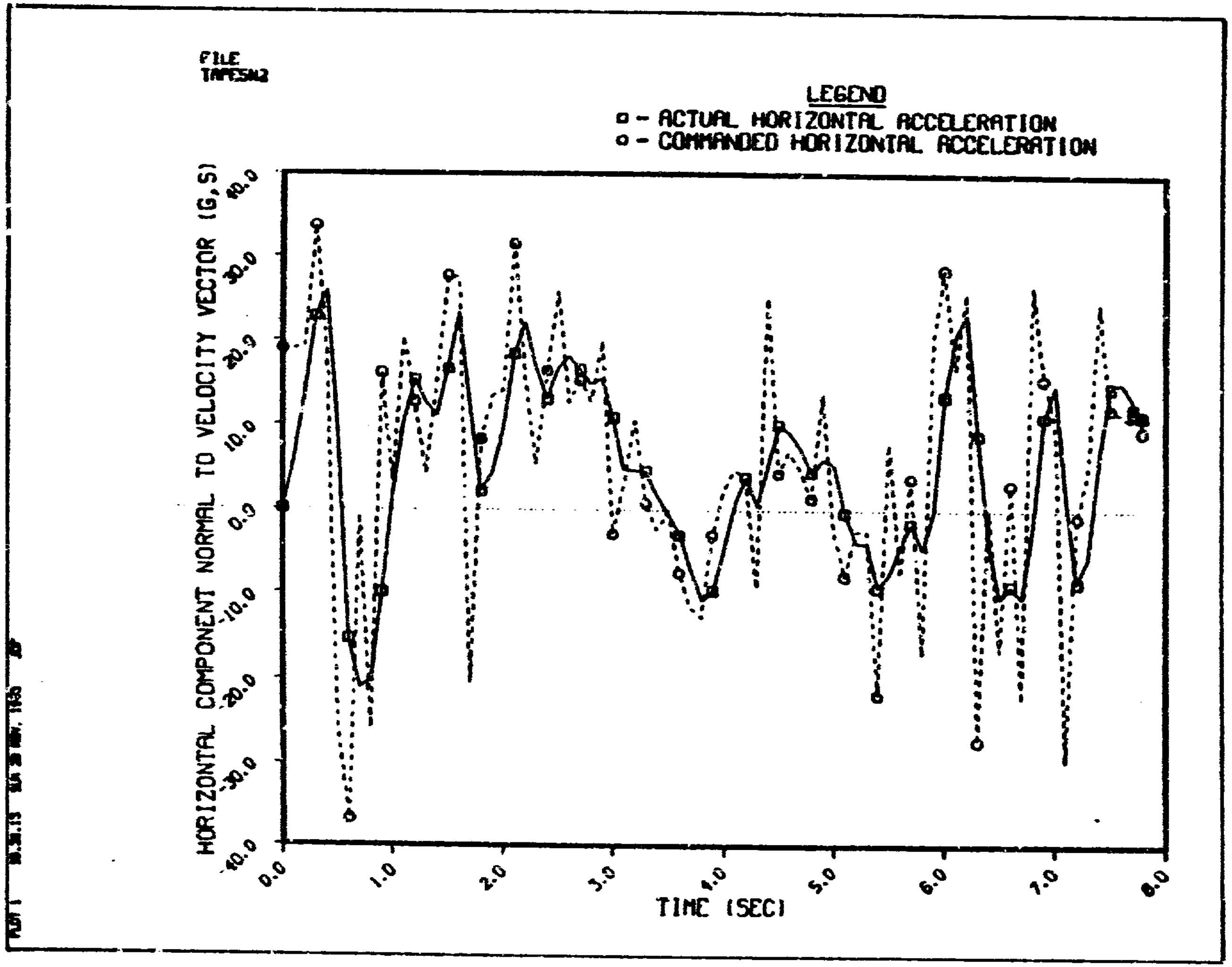


Figure E-42. Acoma, Tail Attack, Turning Tgt, Stochastic, DT = .1 sec

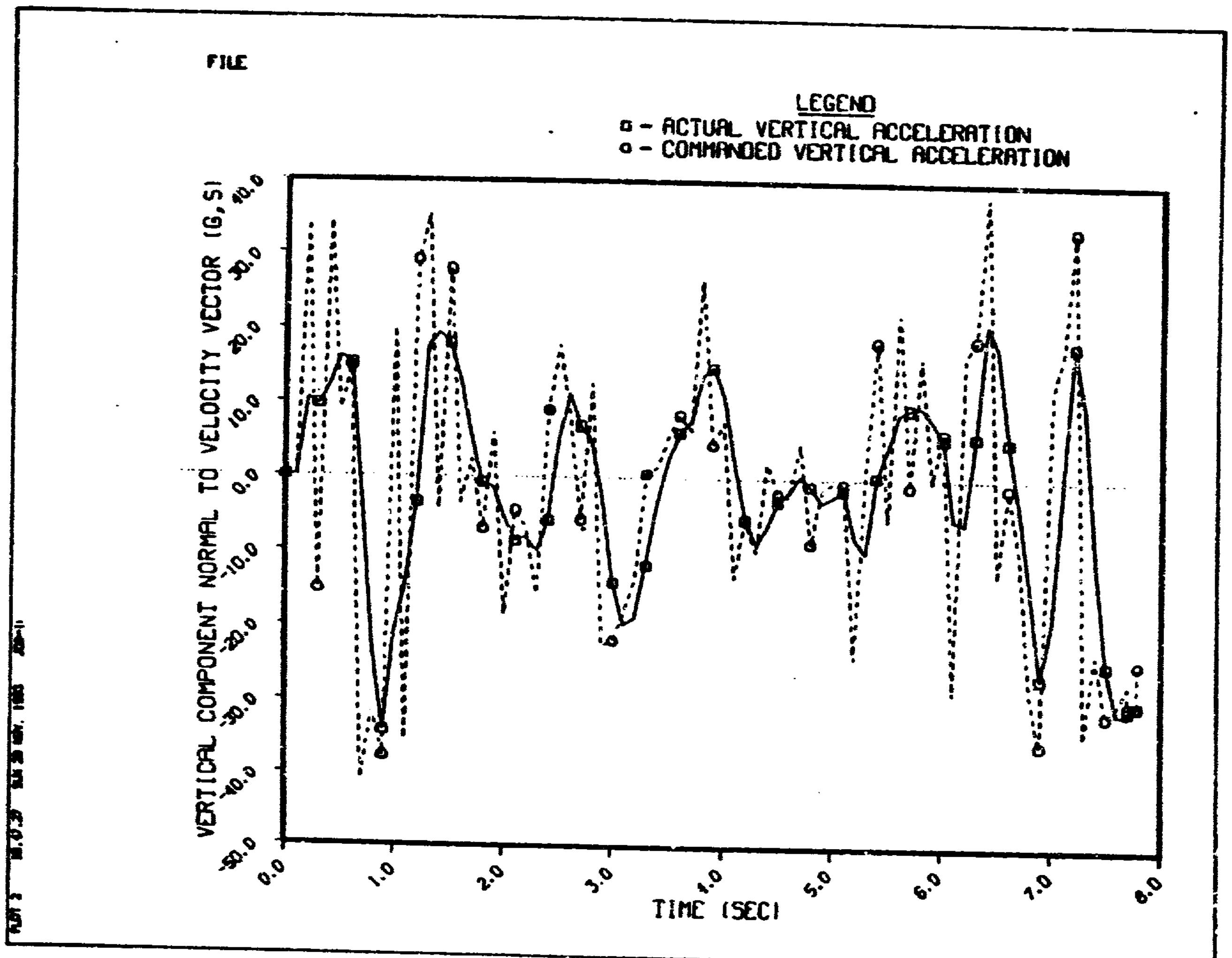


Figure E-43. Acomd, Tail Attack, Turning Tgt, Stochastic, DT = .1 sec

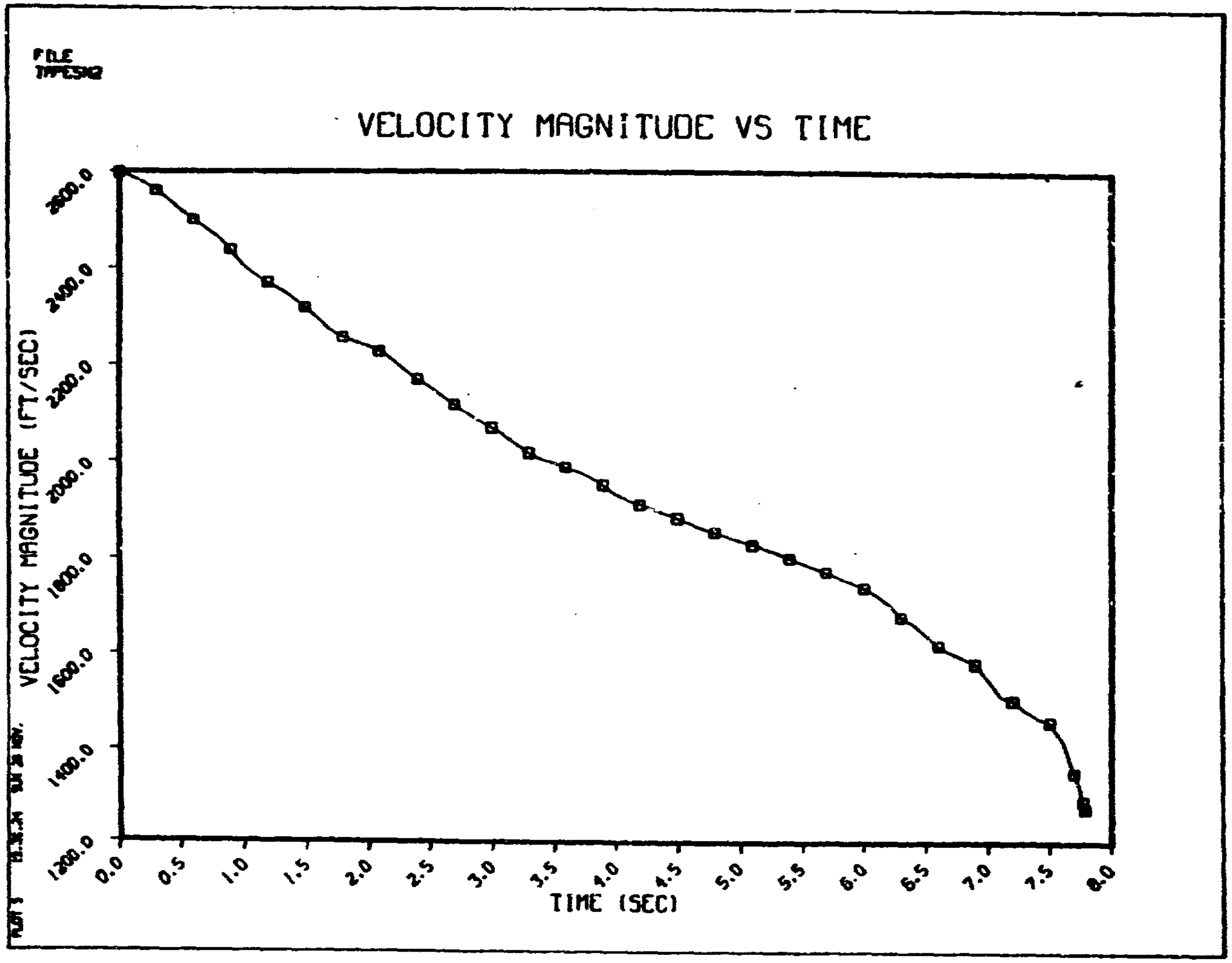


Figure E-44. Velocity vs Time, Tail Attack, Turning Tgt, Stochastic, DT = .1 sec

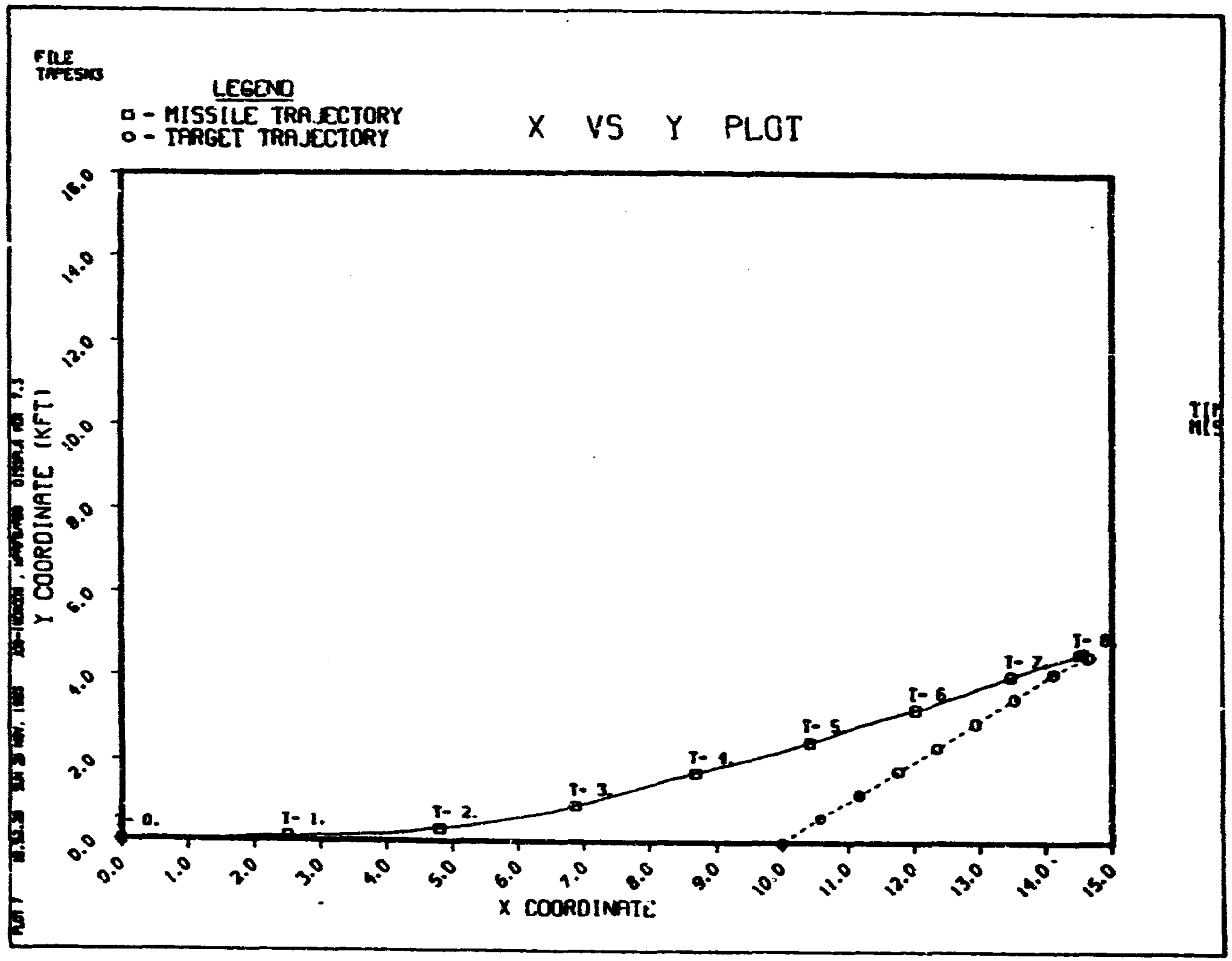


Figure E-45. X Y, Tail Attack, Climb/Dive Tgt, Stochastic, DT = .1 sec

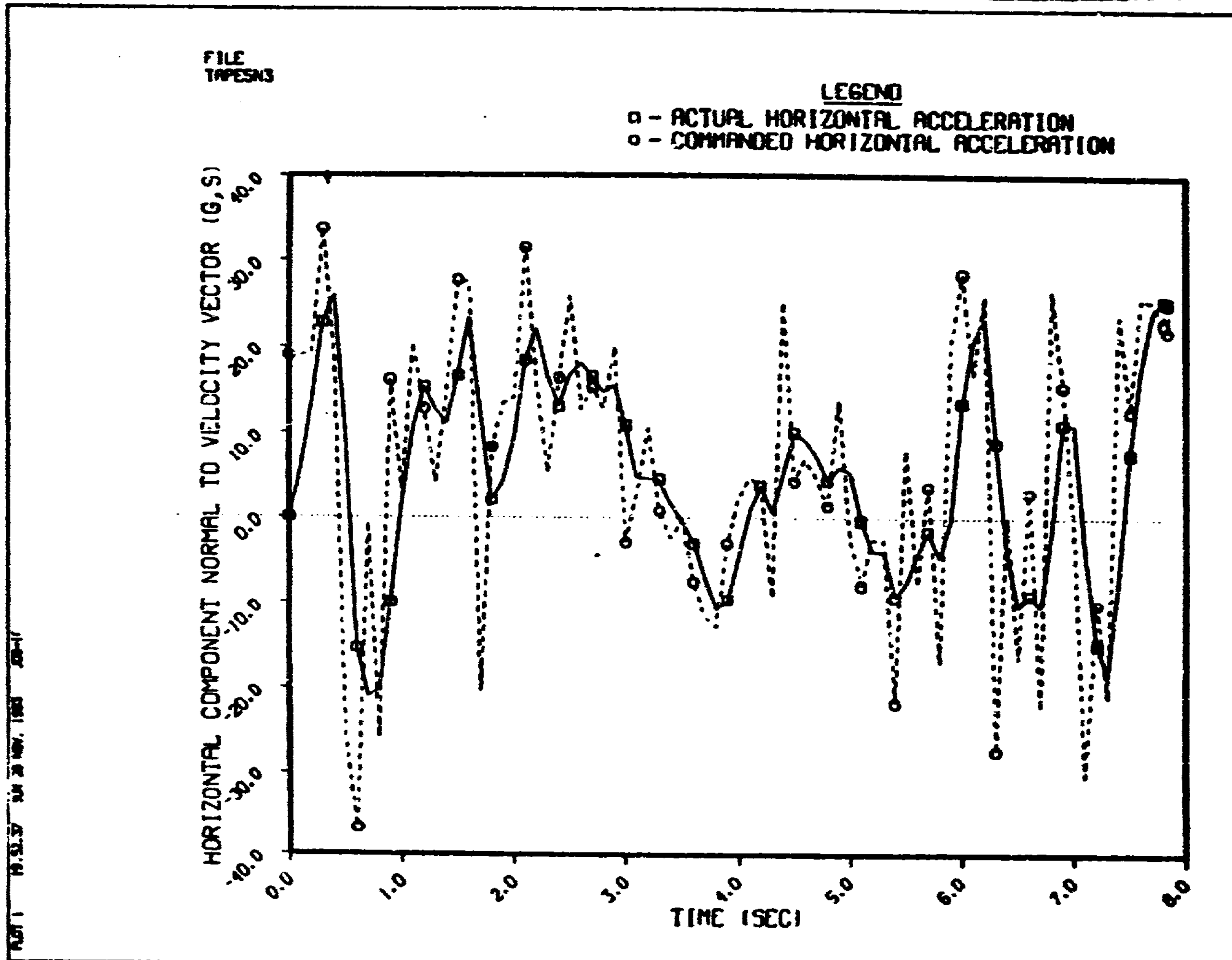


Figure E-46. Acoma, Tail Attack, Climb/Dive Tgt, Stochastic, DT = .1 sec

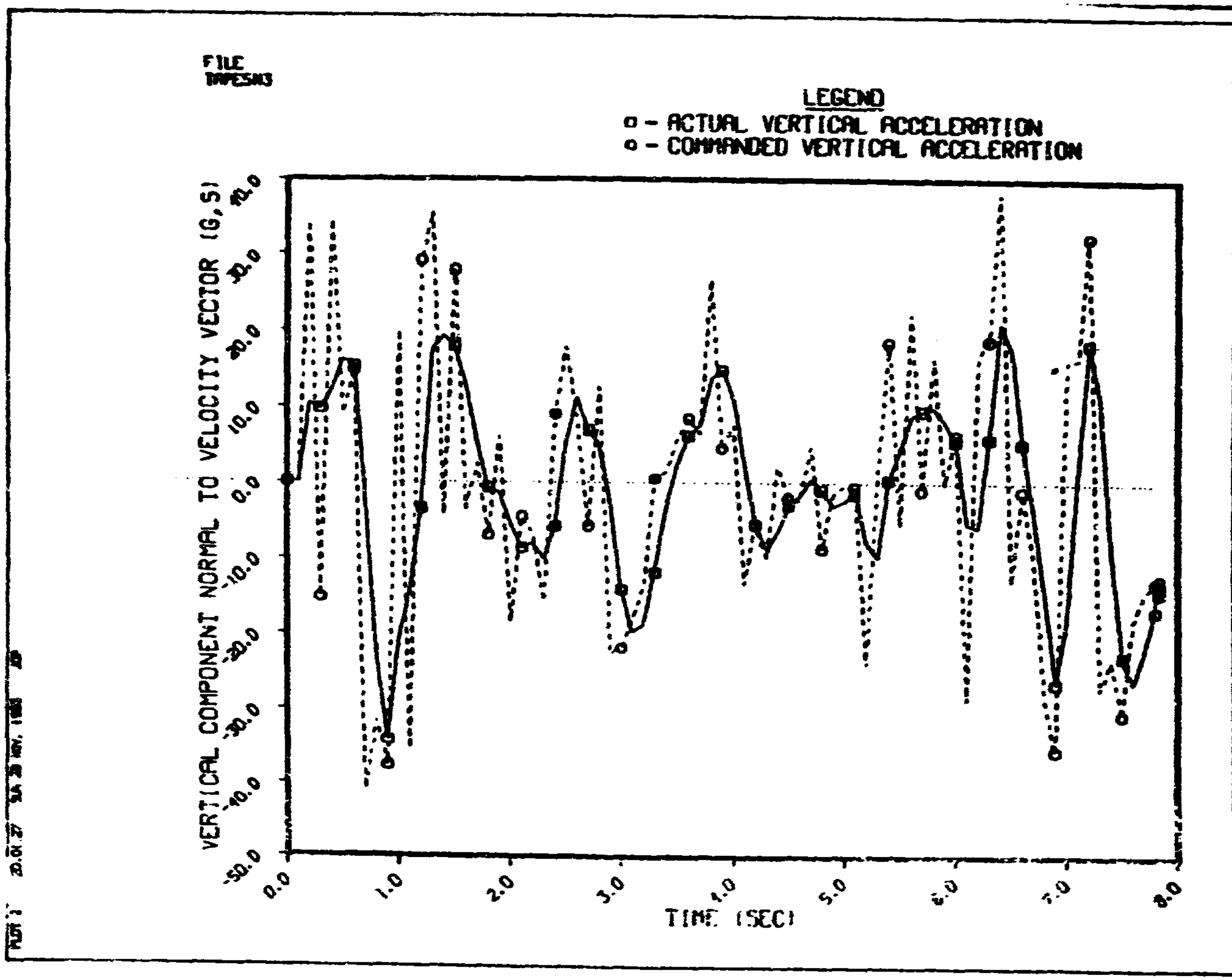


Figure E-47. Acomd, Tail Attack, Climb/Dive Tgt, Stochastic, DT = .1 sec

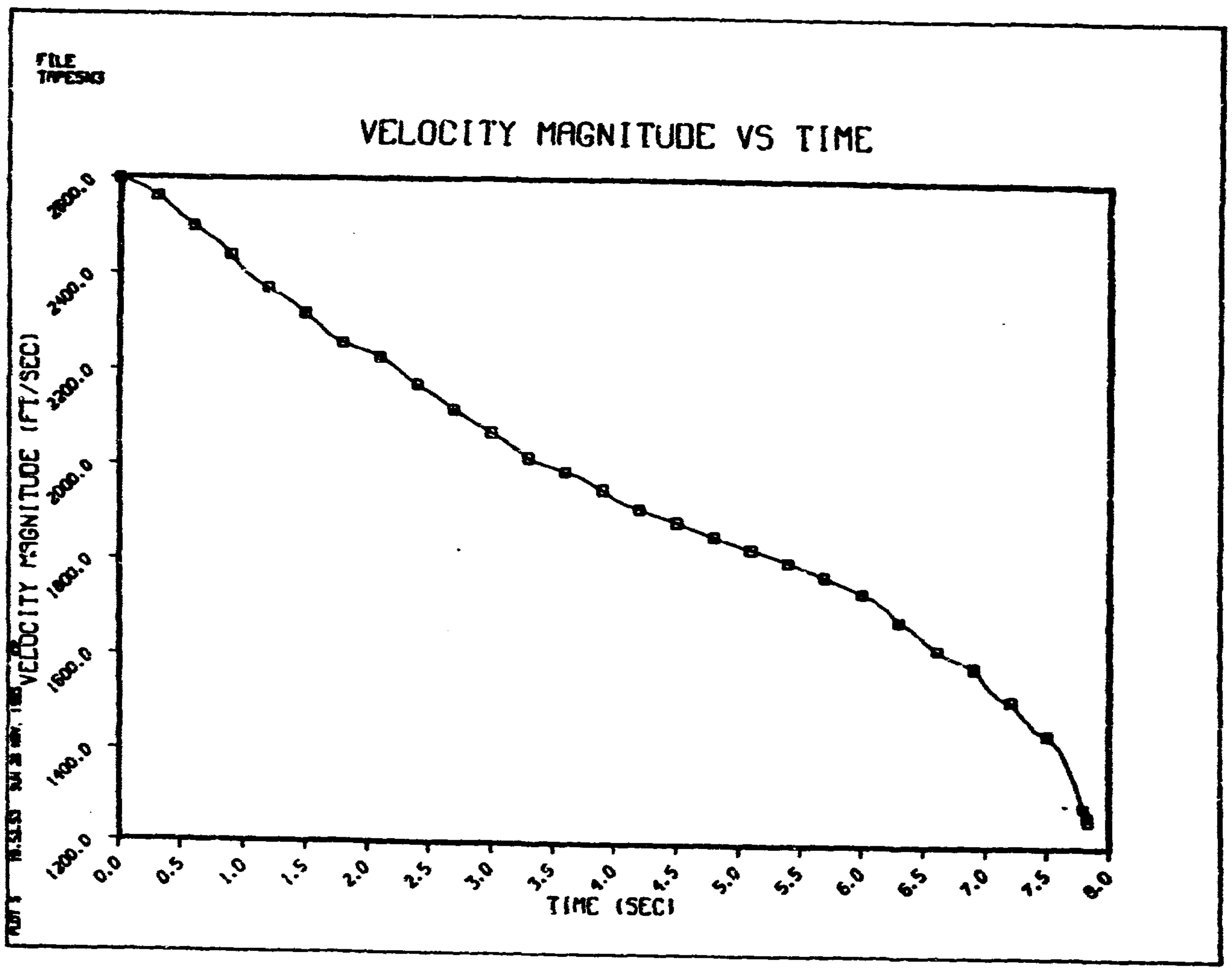


Figure E-48. Velocity vs Time, Tail Attack, Climb/Dive Tgt, Stochastic, DT = .1 sec

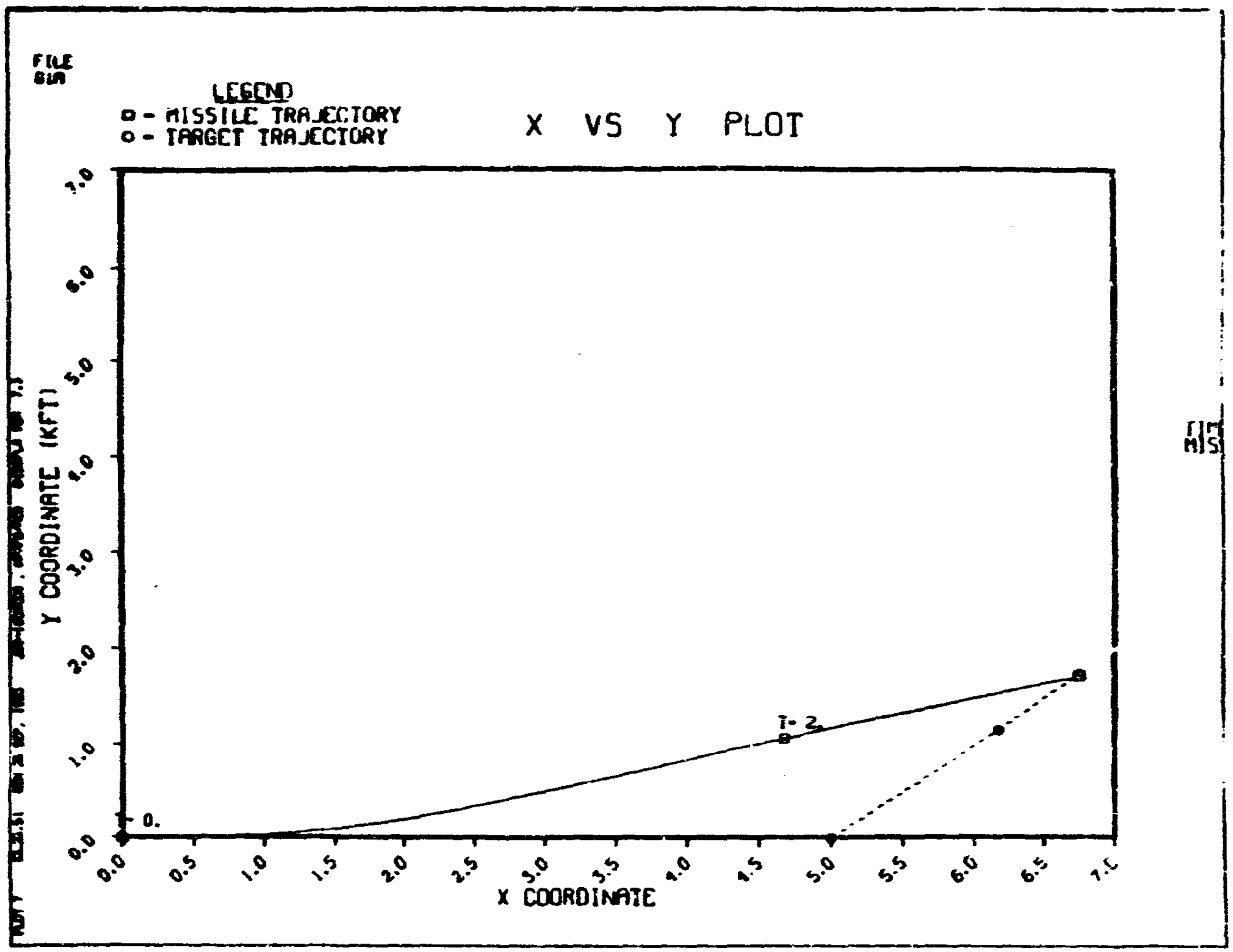


Figure E-49. X Y, Tail Attack, Str/Lvl Tgt, Range = 5K, Deterministic, DT = .01 sec

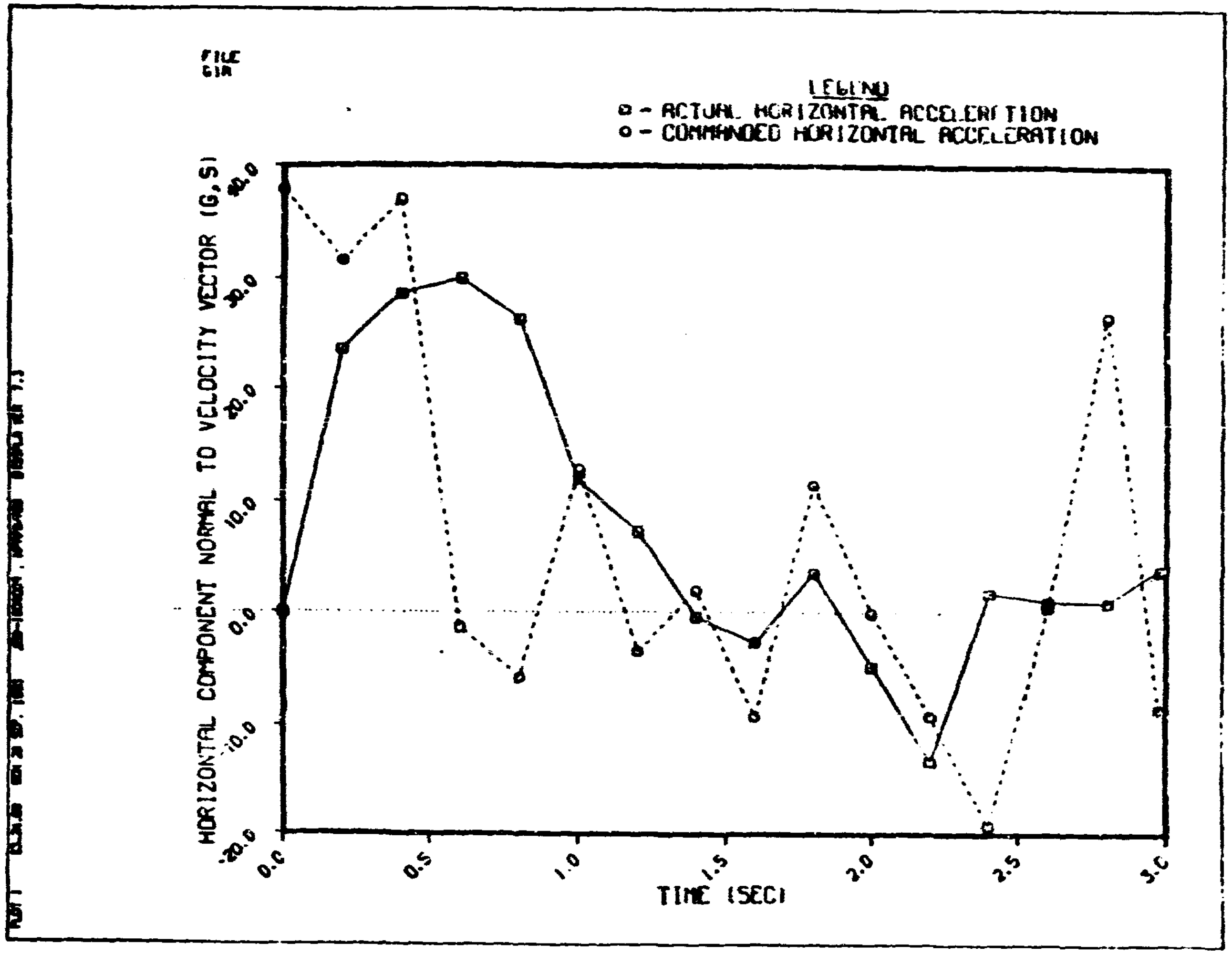


Figure E-50. Acoma, Tail Attack, Str/Lvl Tgt, Range = 5K, Deterministic, DT = .01 sec

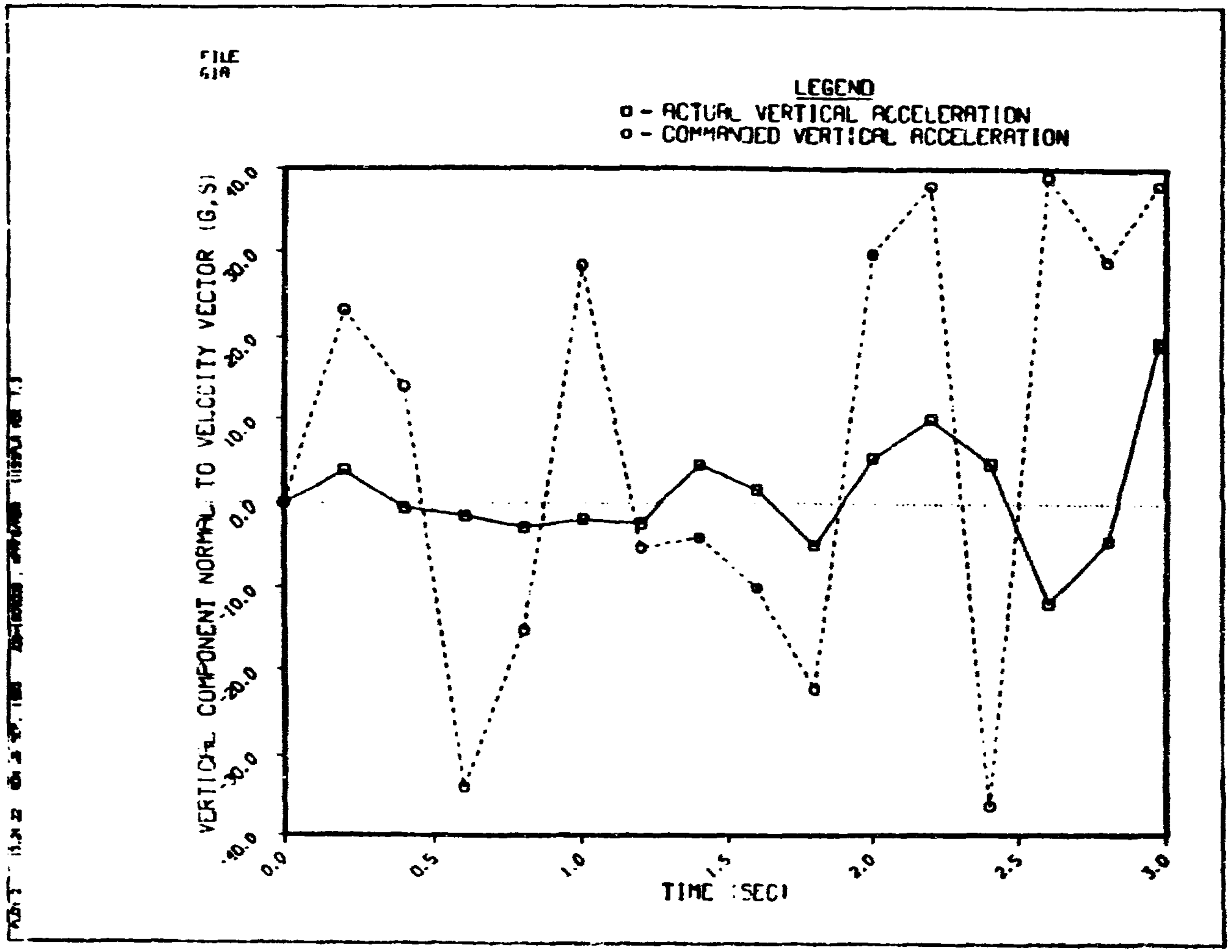


Figure E-51. Acomd, Tail Attack, Str/Lvl Tgt, Range = 5K, Deterministic, DT = .01 sec

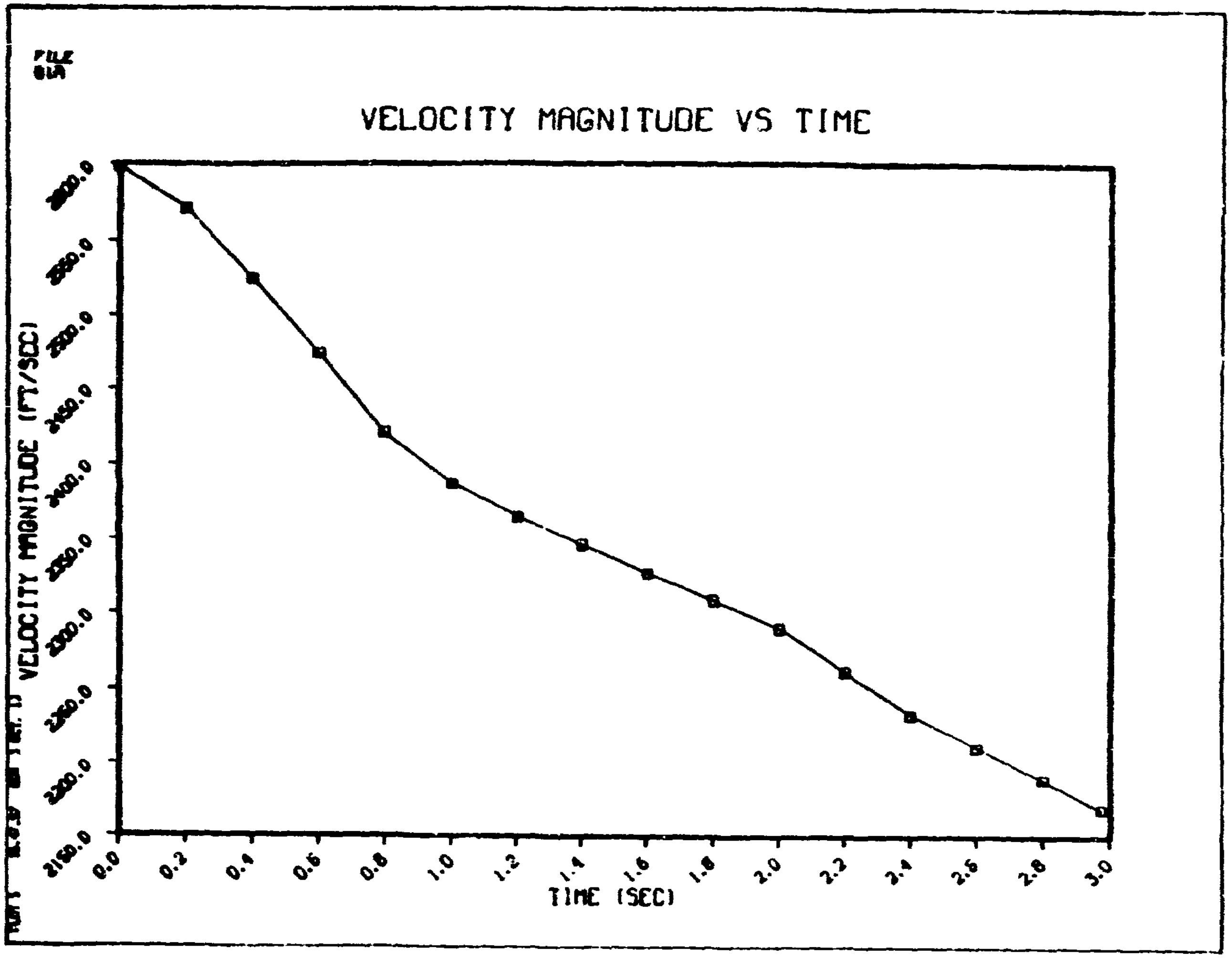


Figure E-52. Velocity vs Time, Tail Attack, Str/Lvl Tgt, Range = 5K, Deterministic, DT = .01 sec

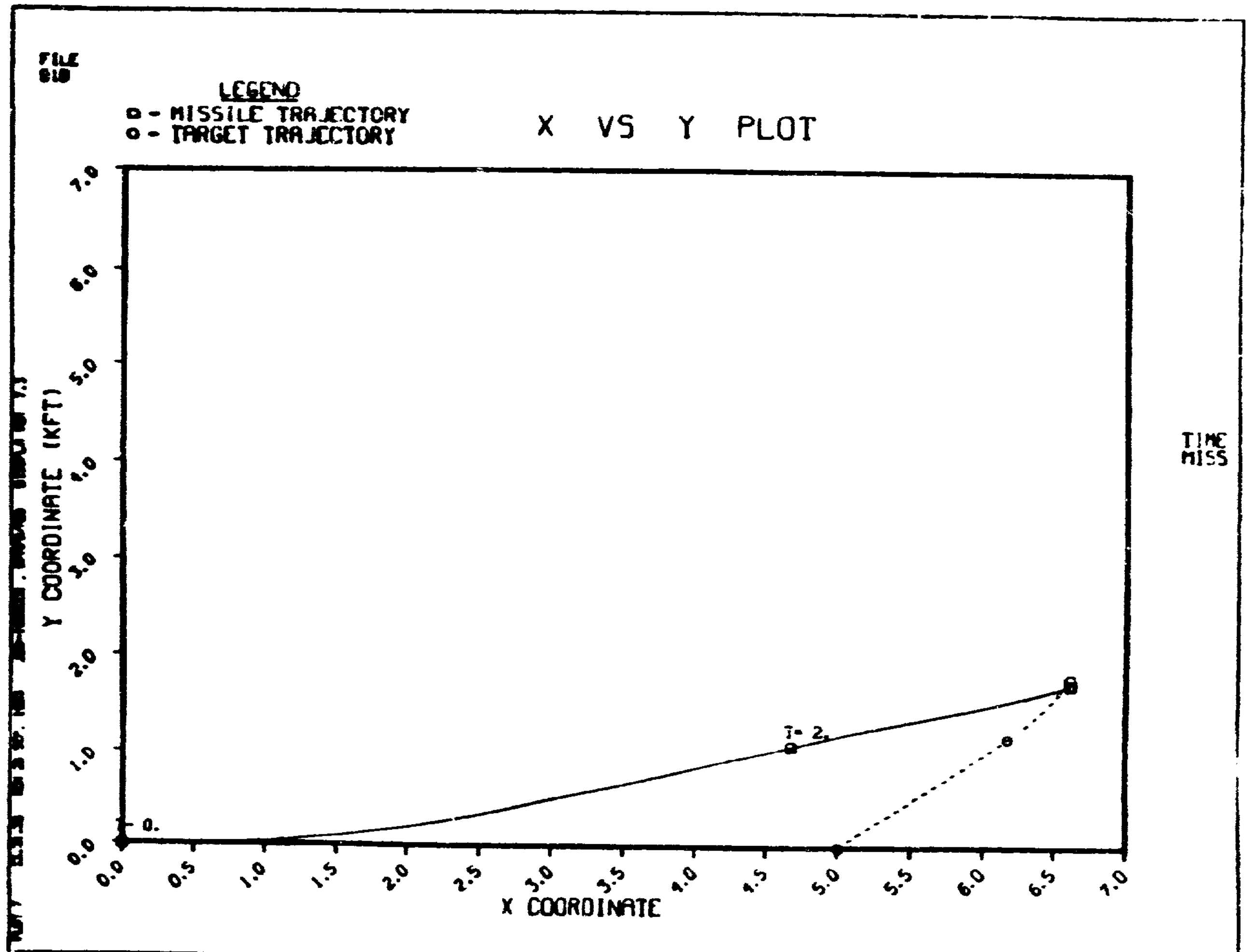


Figure E-53. X Y, Tail Attack, Turning Tgt, Range = 5K, Deterministic, DT = .01 sec

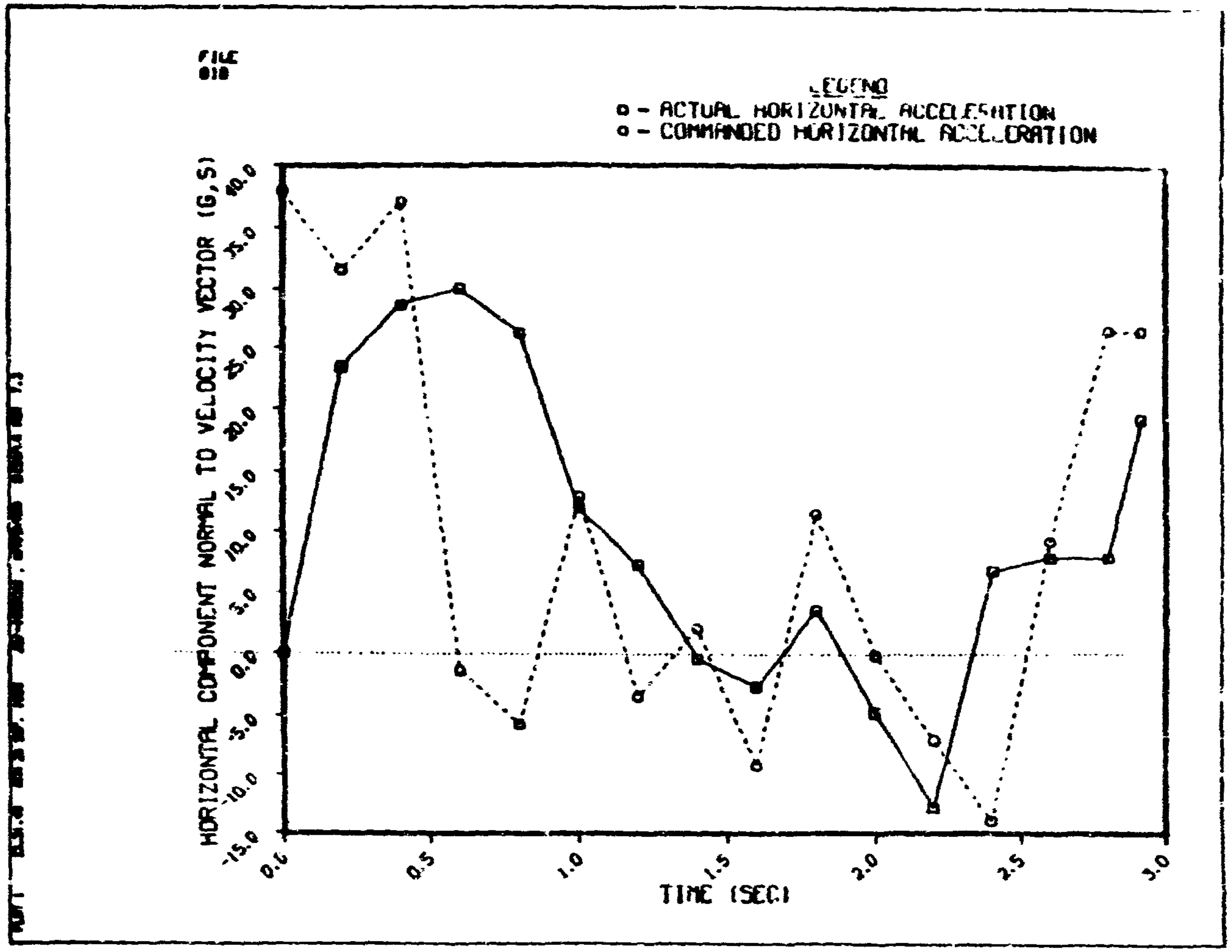


Figure E-54. Accma, Tail Attack, Turning Tgt, Range = 5K, Deterministic, DT = .01 sec

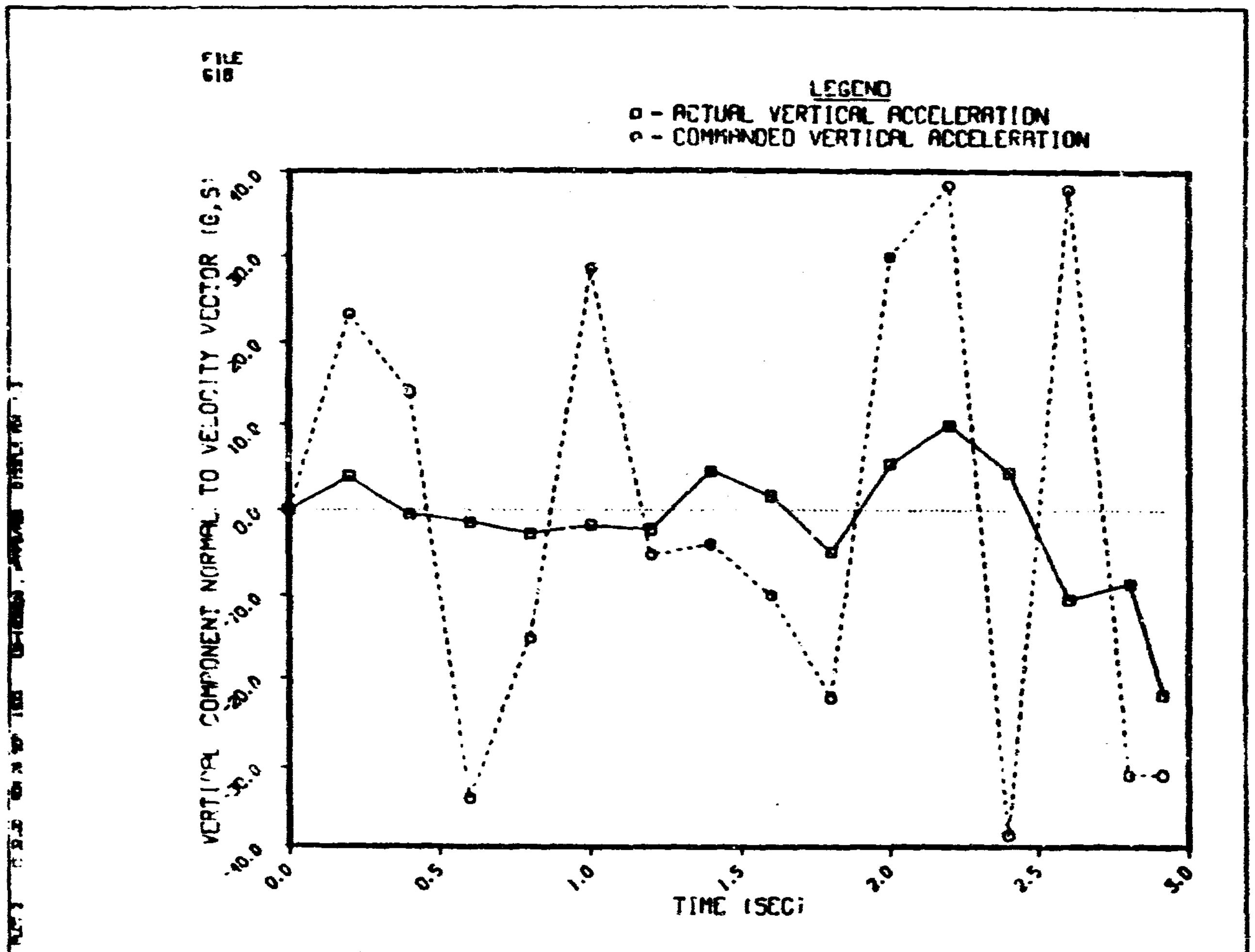


Figure E-55. Acomd, Tail Attack, Turning Tgt, Range = 5K, Deterministic, DT = .01 sec

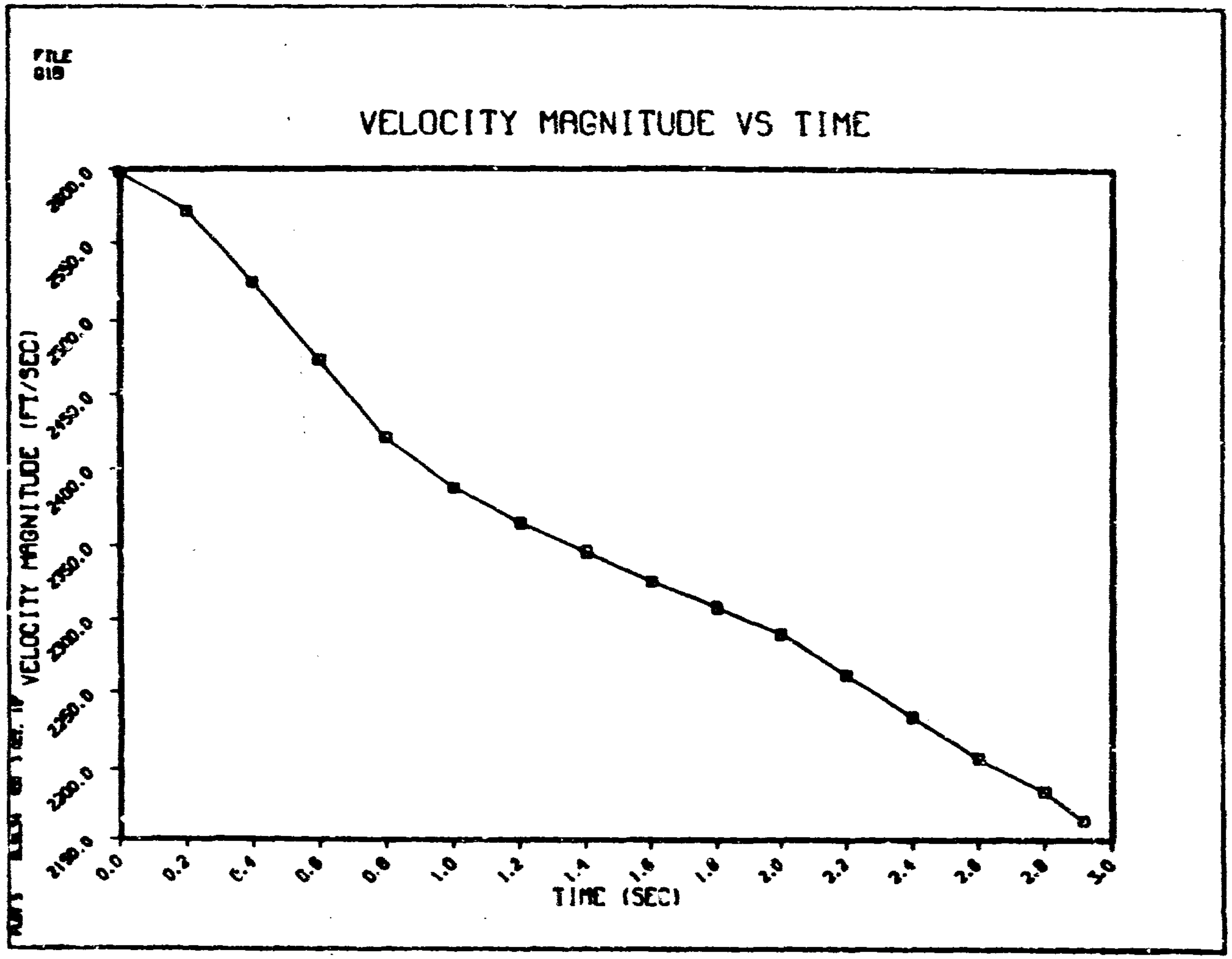


Figure E-56. Velocity vs Time, Tail Attack, Turning Tgt, Range = 5K, Deterministic, DT = .01 sec

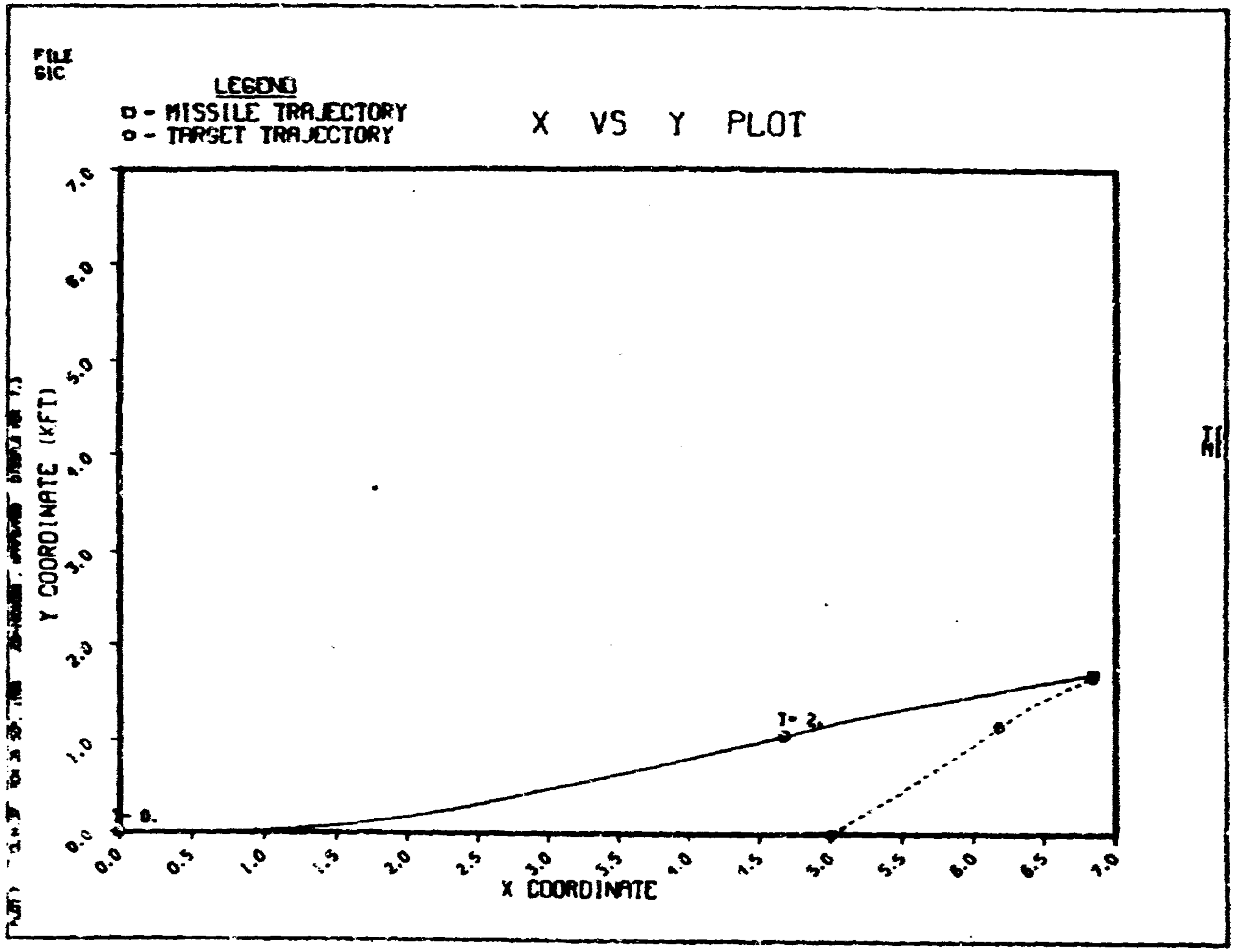


Figure E-57. X Y, Tail Attack, Climb/Dive Tgt, Range = 5K, Deterministic, DT = .01 sec

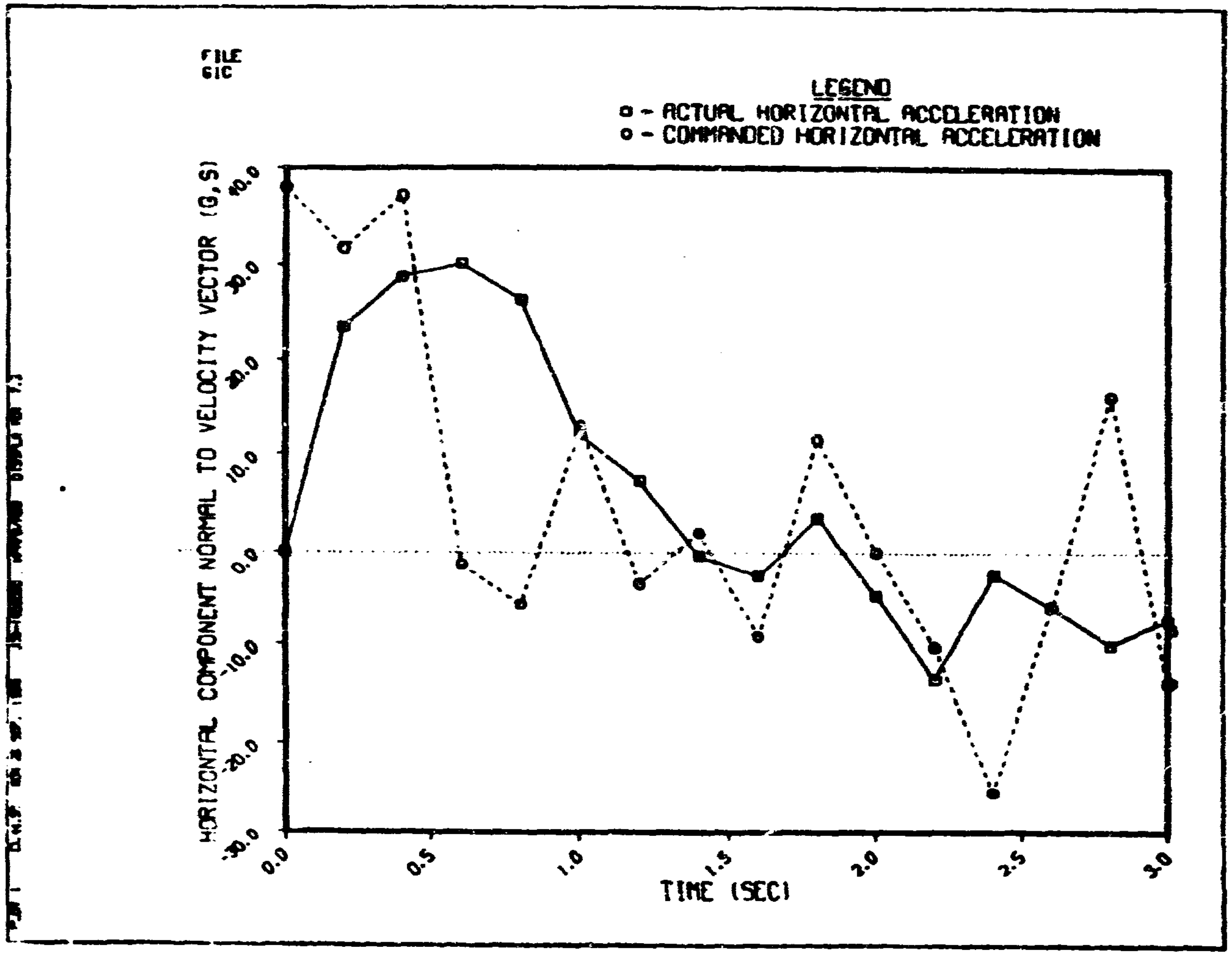


Figure E-58. Acoma, Tail Attack, Climb/Dive Tgt, Range = 5K, Deterministic, DT = .01 sec

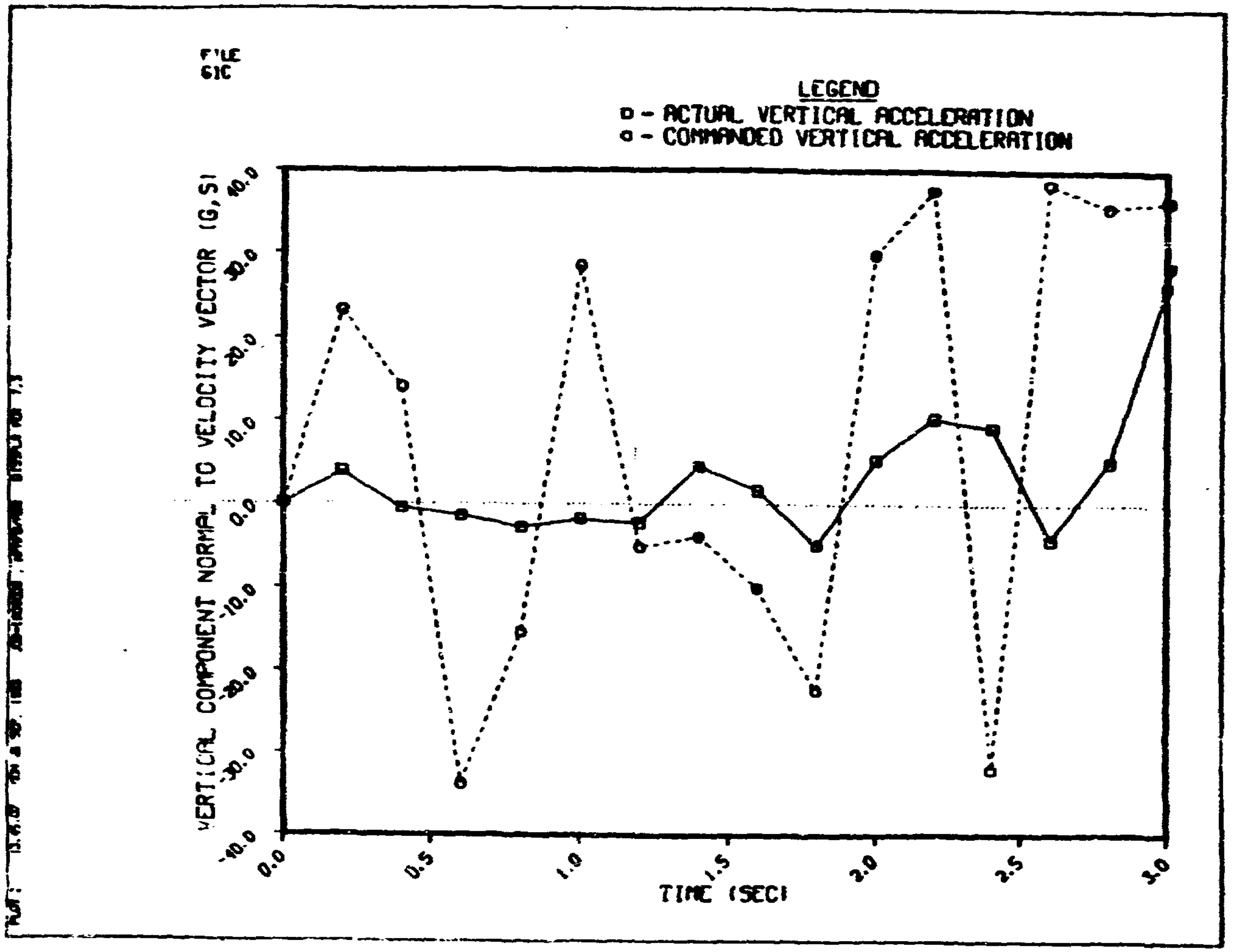


Figure E-59. Acomd, Tail Attack, Climb/Dive Tgt, Range = 5K, Deterministic, DT = .01 sec

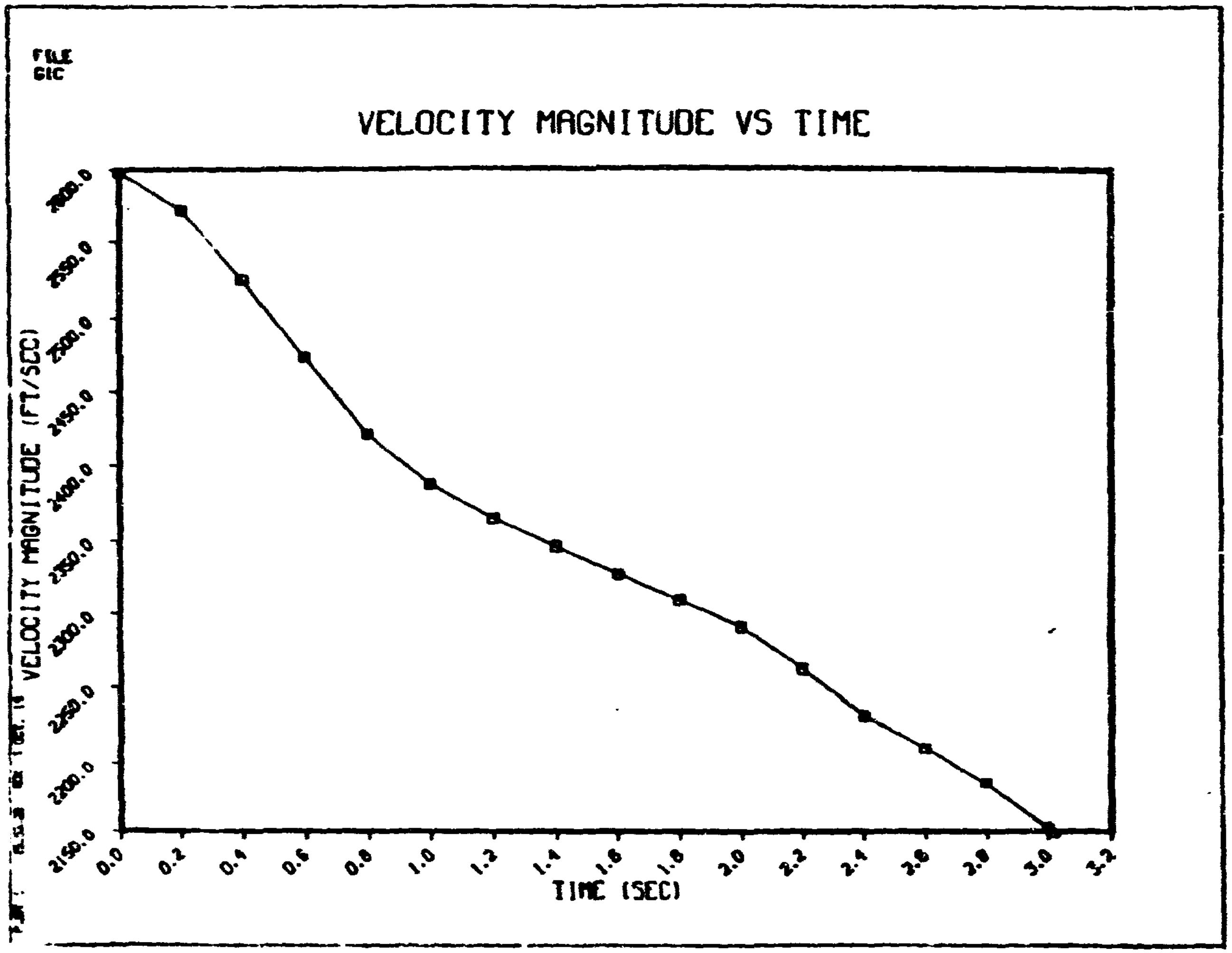


Figure E-60. Velocity vs Time, Climb/Dive Tgt, Range = 5K, Deterministic, DT = .01 sec

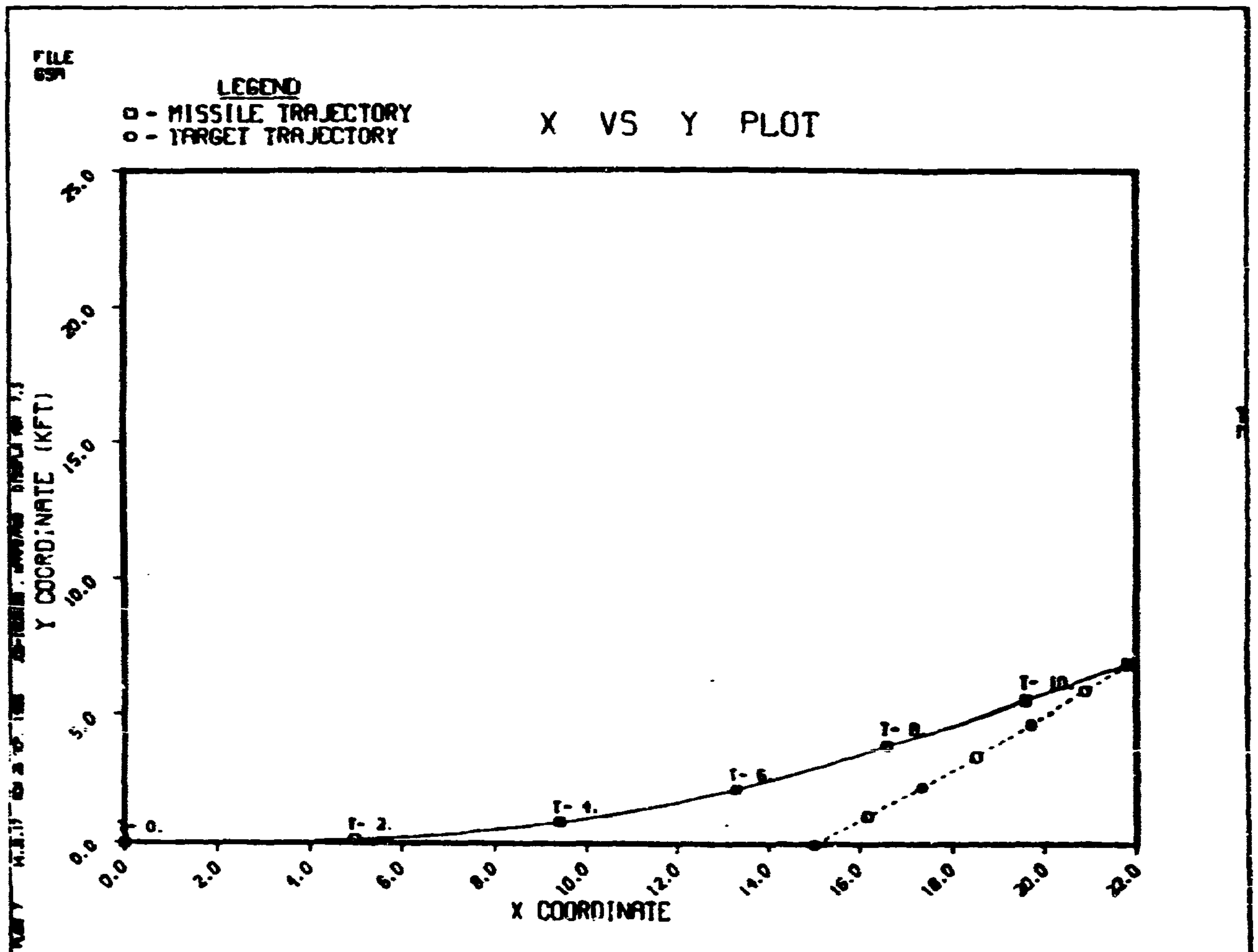


Figure E-61. X Y, Tail Attack, Str/Lvl Tgt, Range = 15K, Deterministic, DT = .01 sec

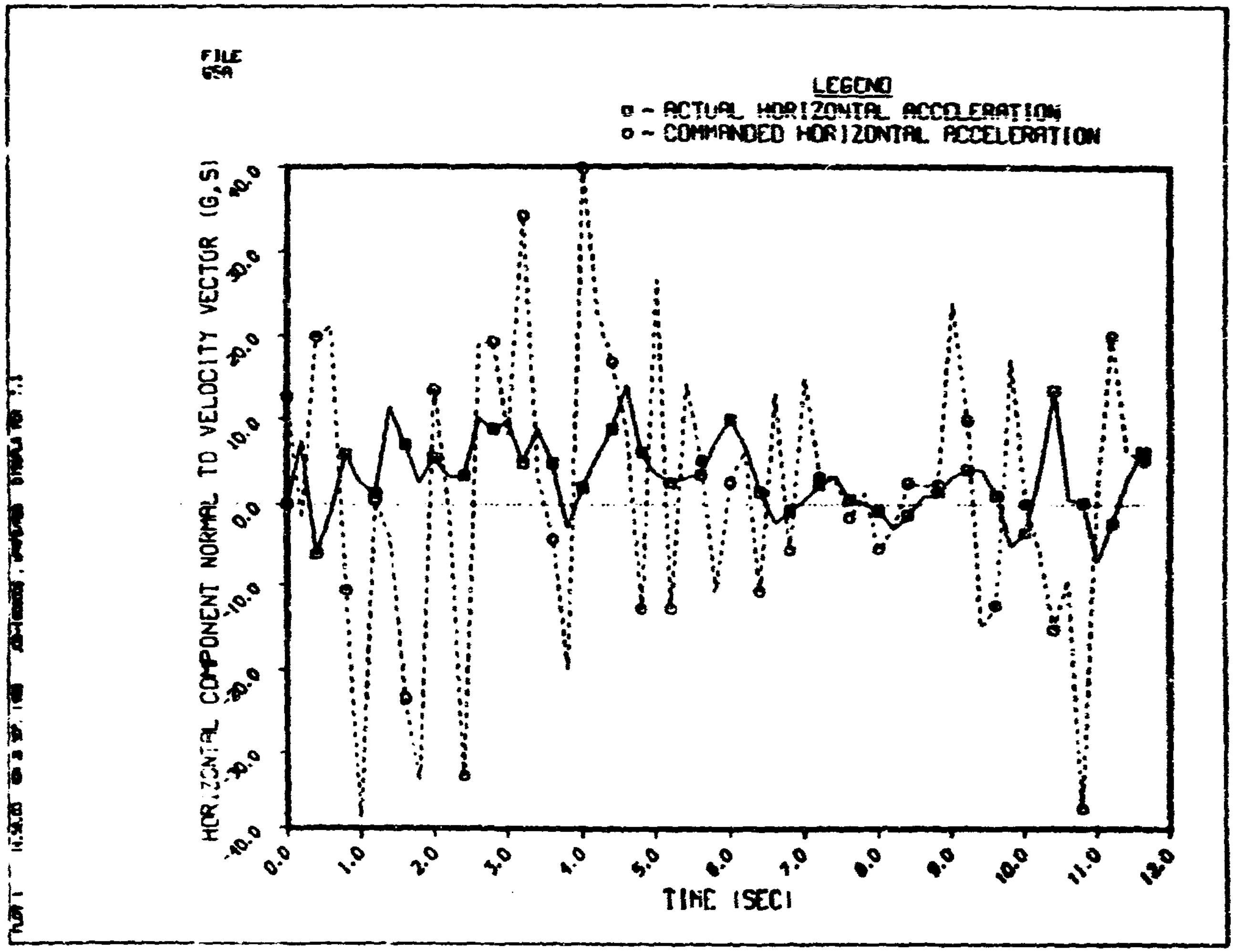


Figure E-62. Acoma, Tail Attack, Str/Lvl Tgt, Range = 15K, Deterministic, DT = .01 sec

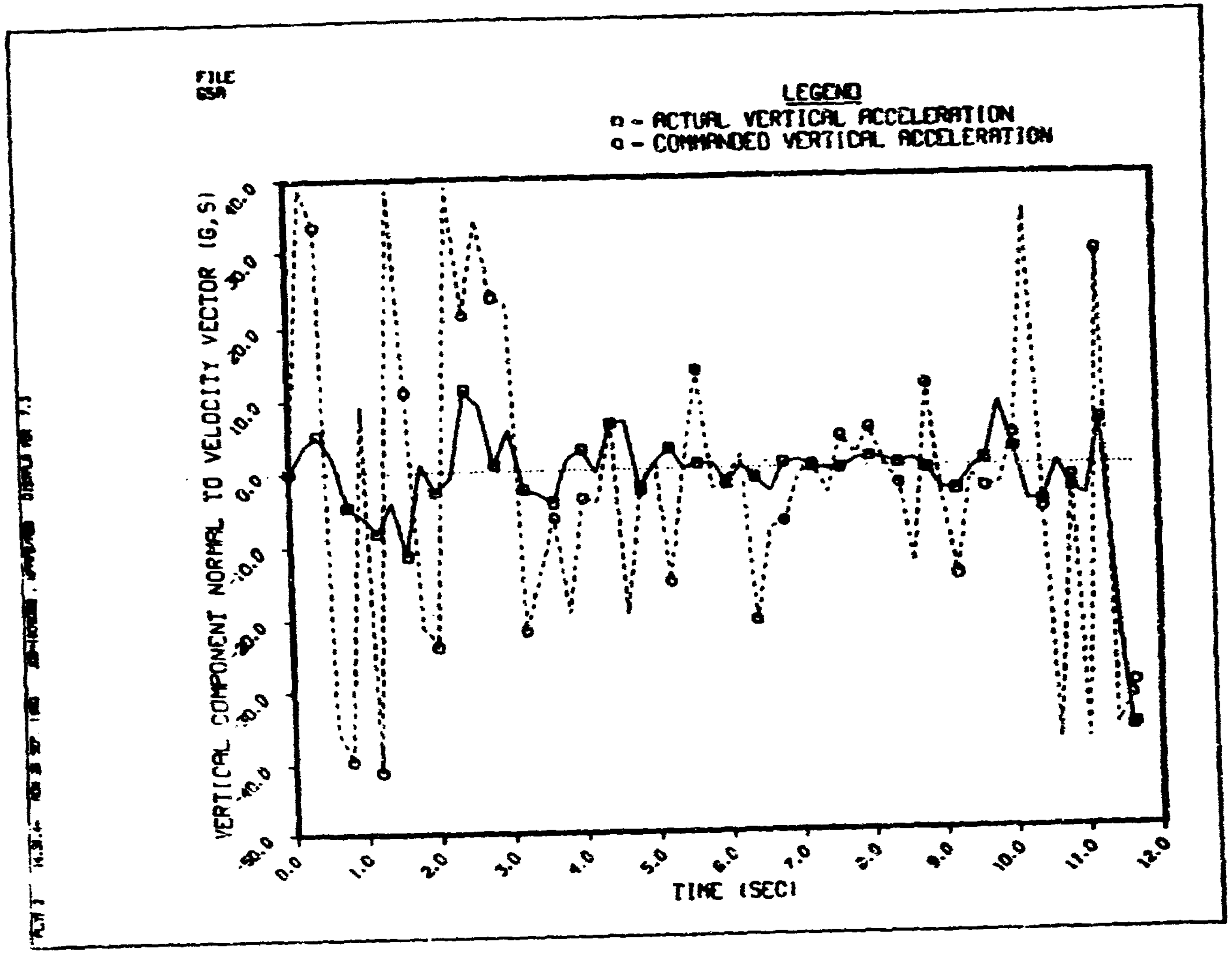


Figure E-63. Acomd, Tail Attack, Str/Lvl Tgt, Range = 15K, Deterministic, DT = .01 sec

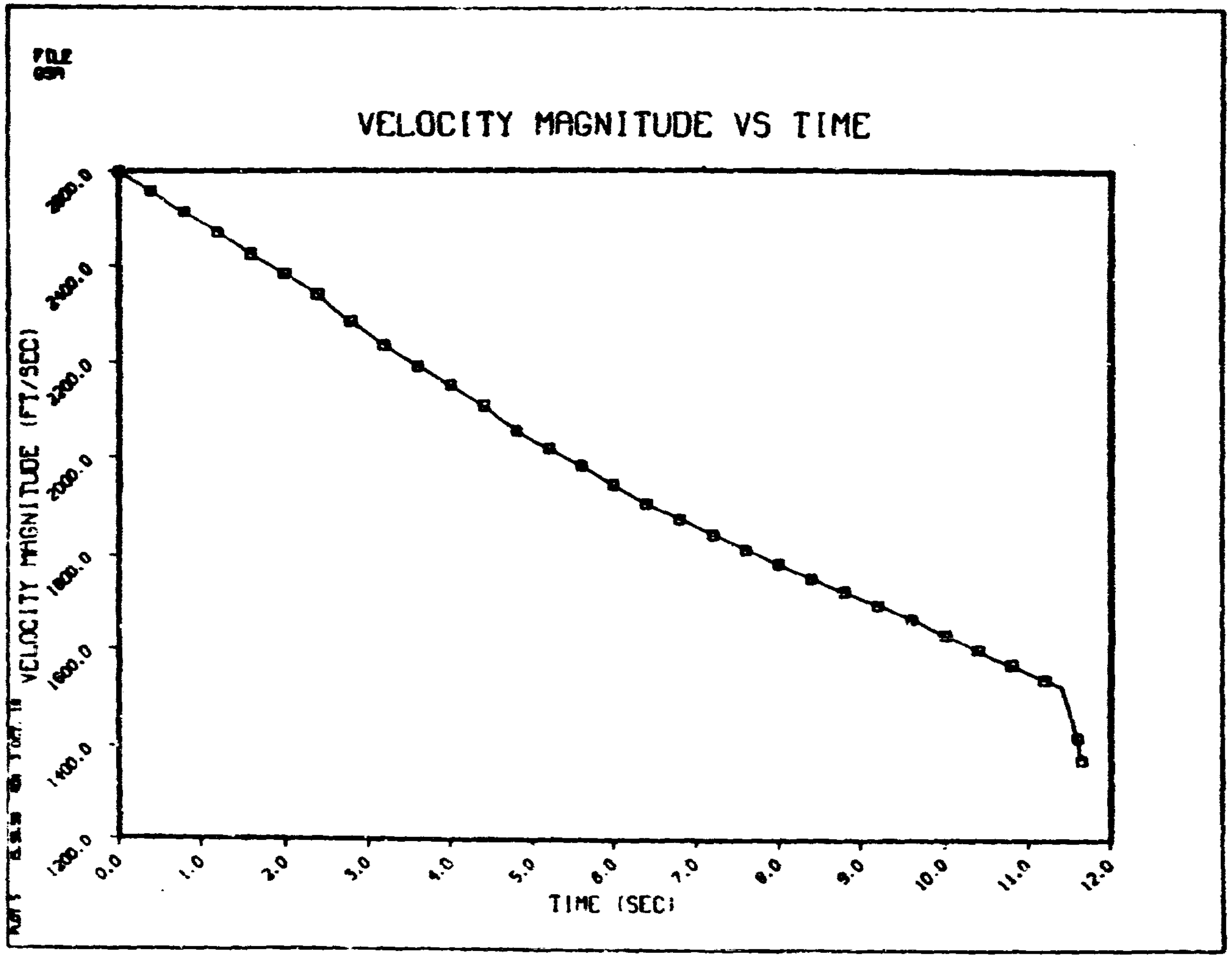
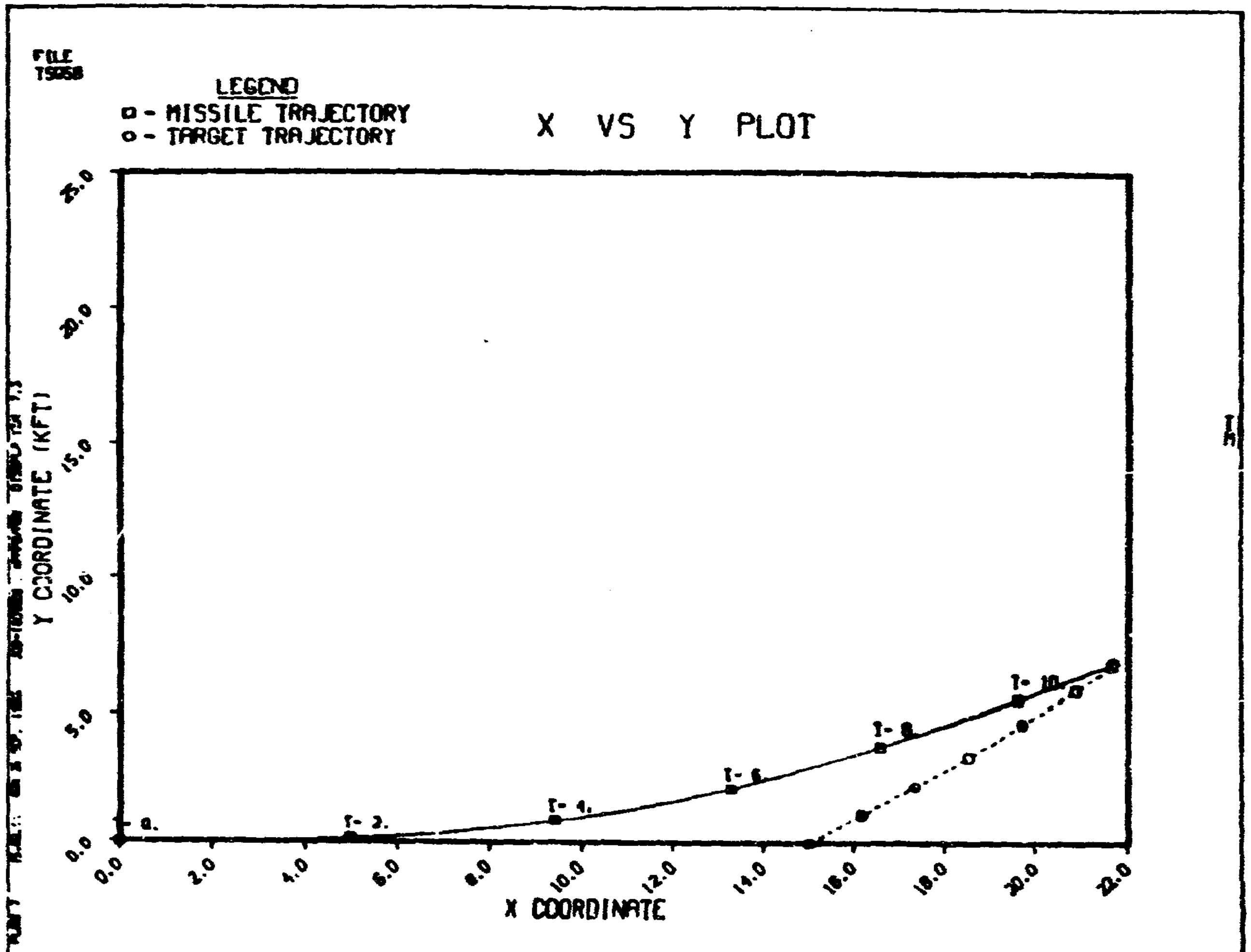


Figure E-64. Velocity vs Time, Tail Attack, Str/Lvl Tgt, Range = 25K, Deterministic, DT = .01 sec



Fiugre E-65. X Y, Tail Attack, Turning Tgt, Range = 15K, Deterministic, DT = .01 sec

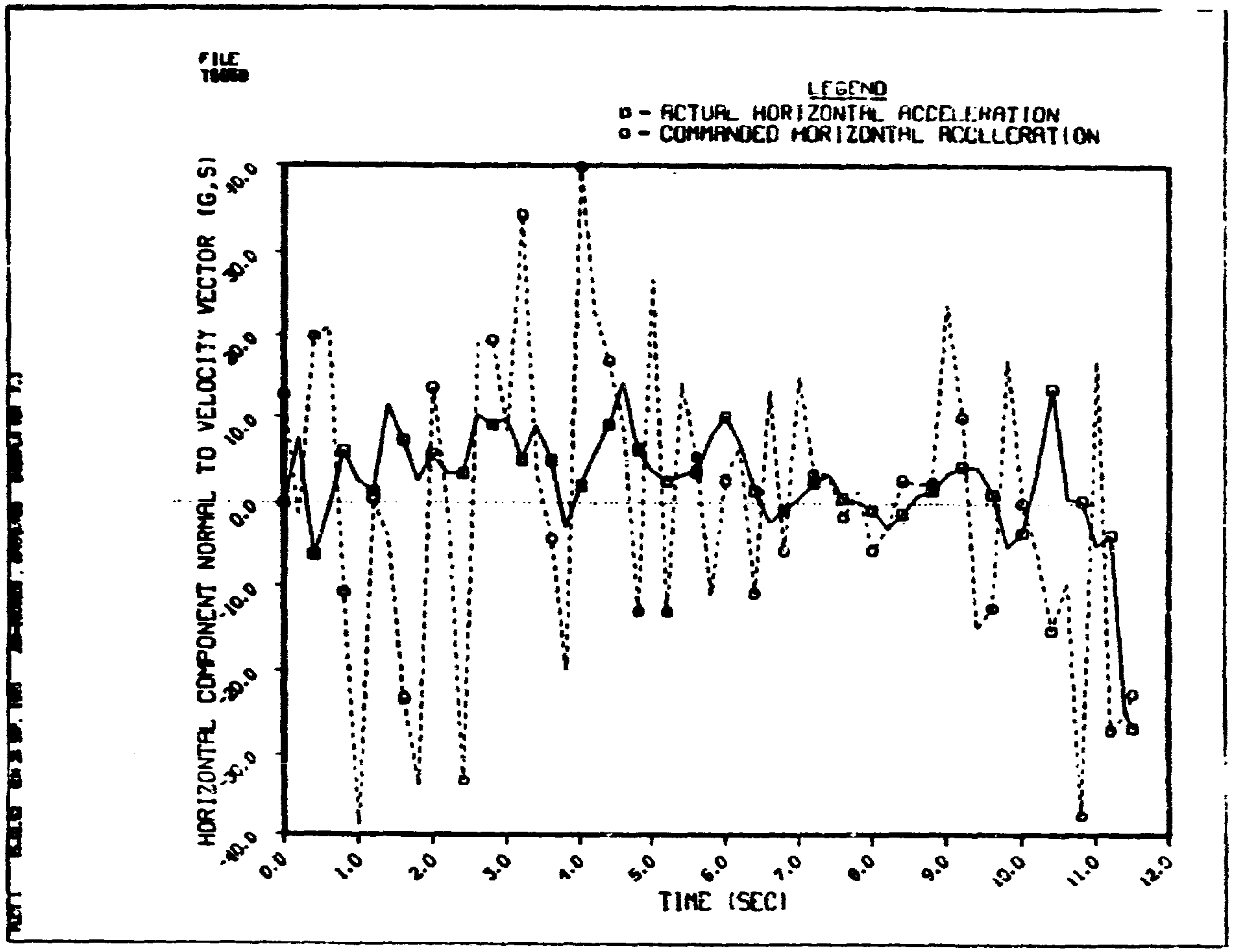


Figure E-66. Acoma, Tail Attack, Turning Tgt, Range = 15K, Deterministic, DT = .01 sec

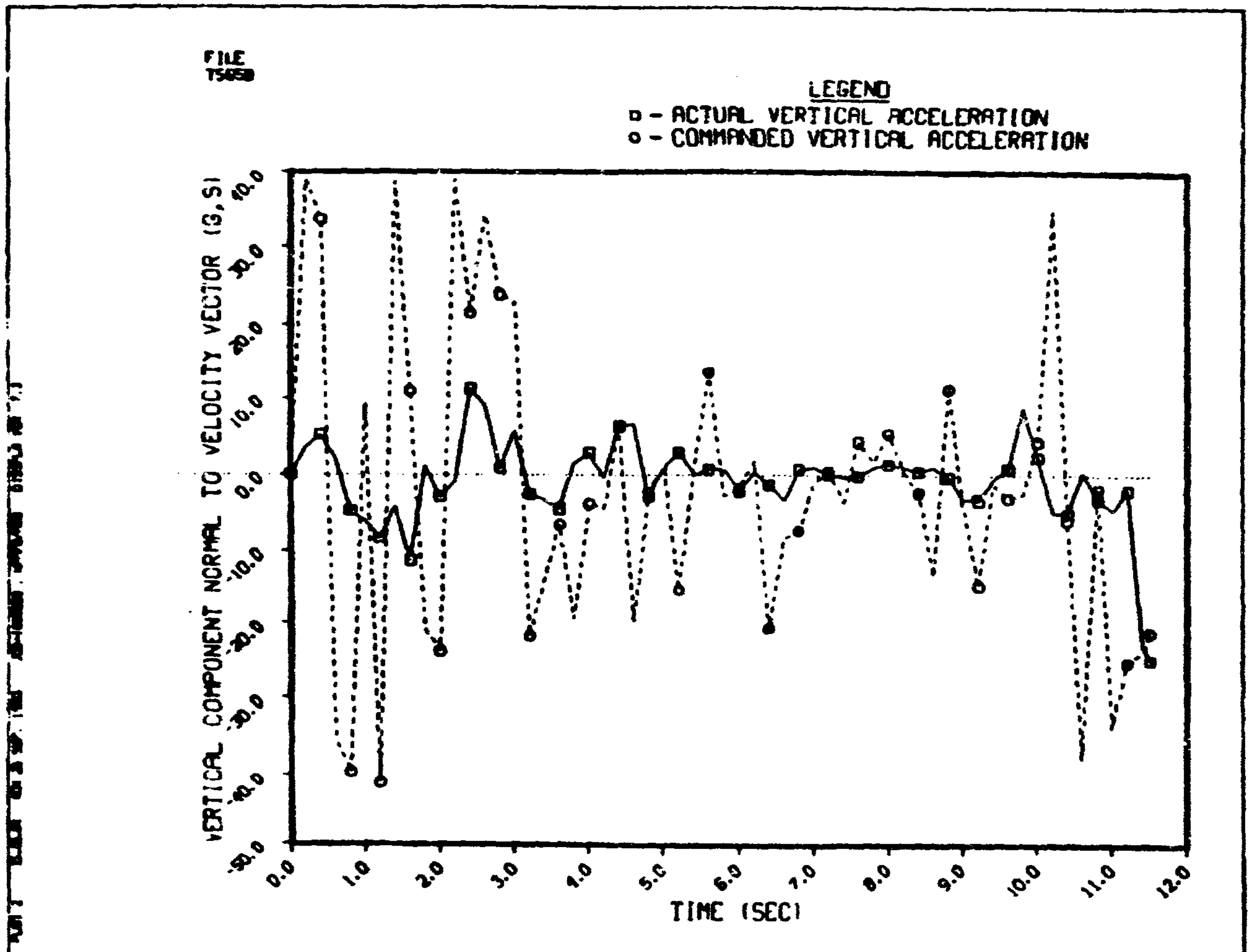


Figure E-67. Acomd, Tail Attack, Turning Tgt, Range = 15K, Deterministic, DT = .01 sec

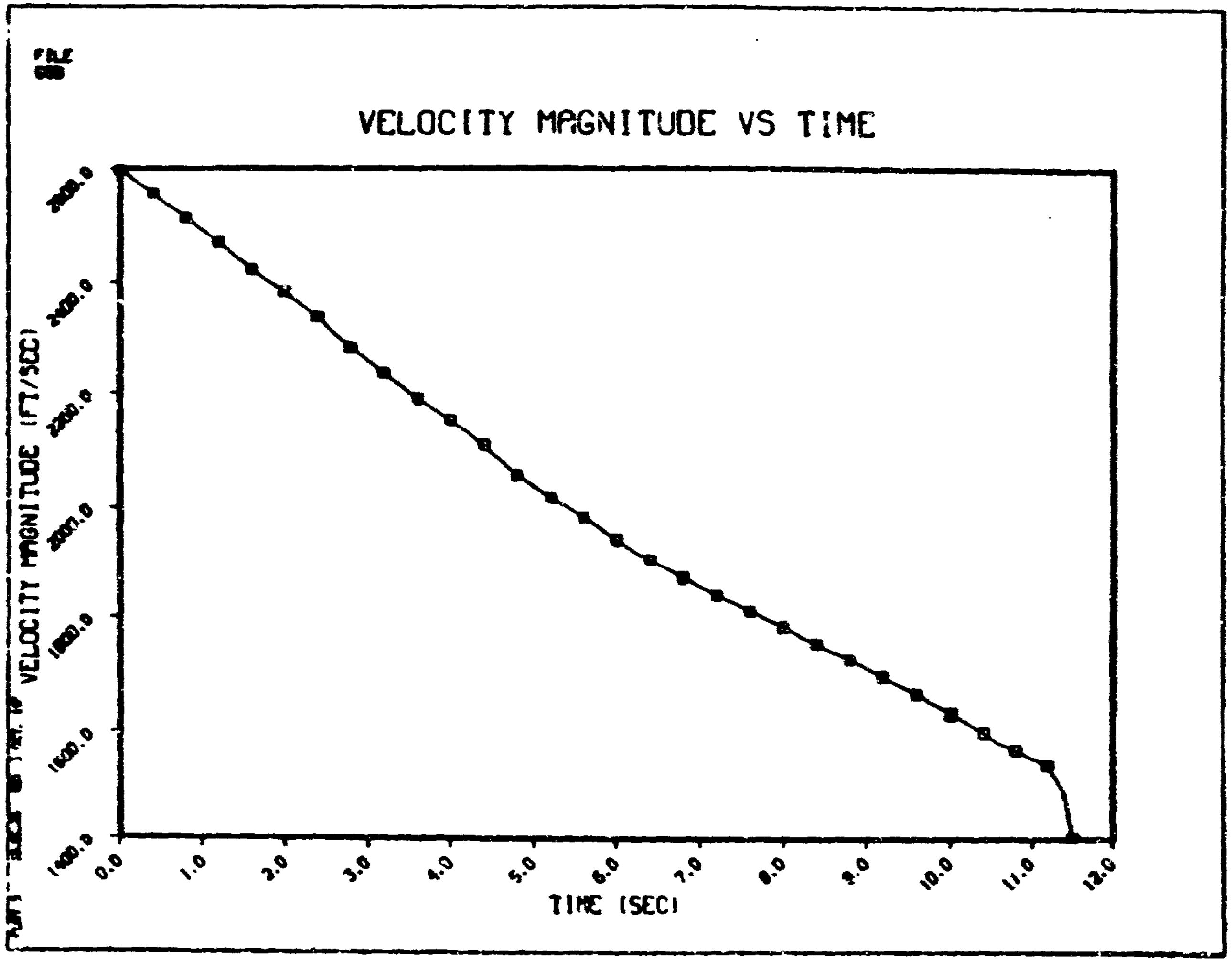


Figure E-68. Velocity vs Time, Tail Attack, Turning Tgt, Range = 15K, Deterministic, DT = .01 sec

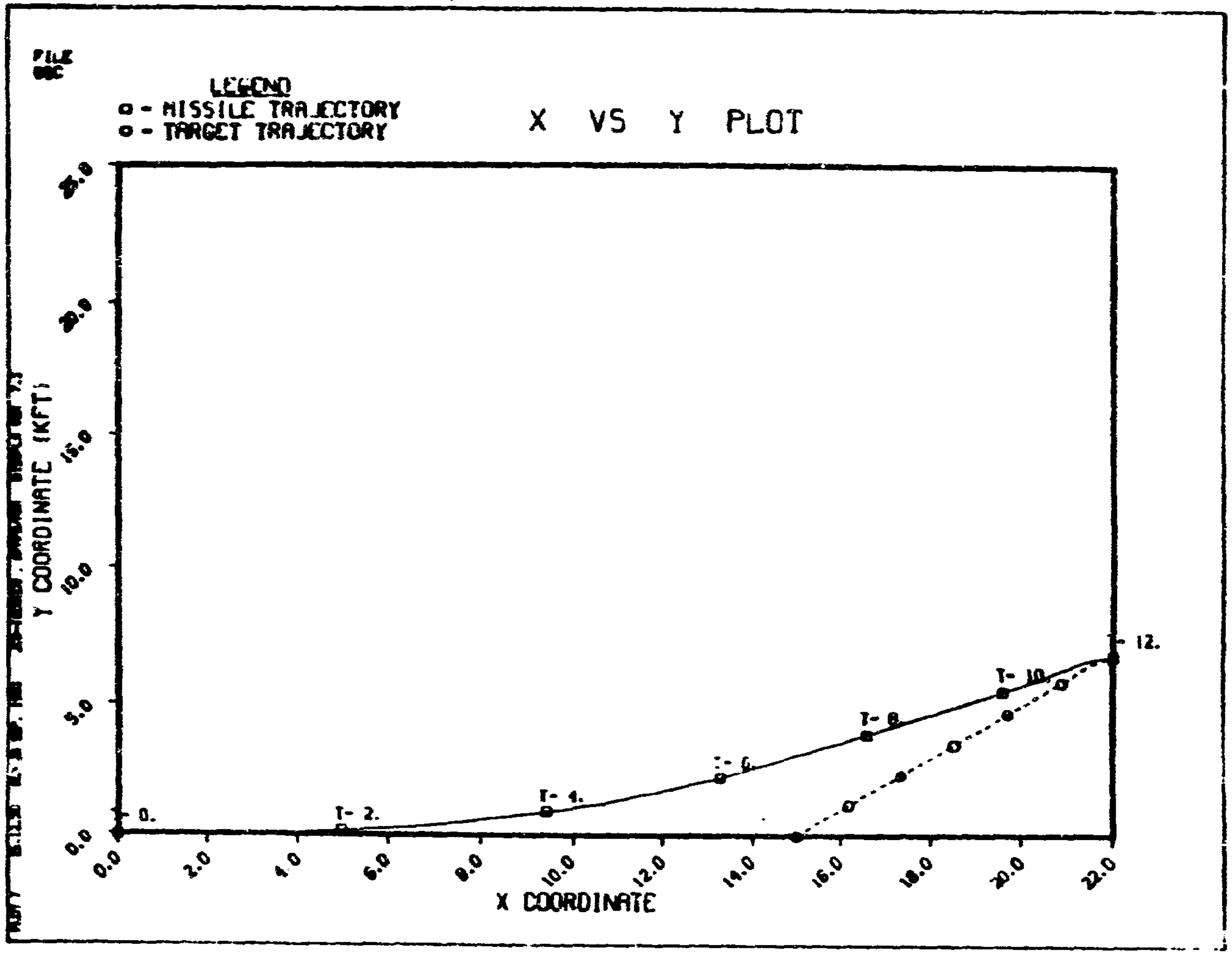


Figure E-69. X Y, Tail Attack, Turning Tgt, Range = 15K, Deterministic, DT = .01 sec

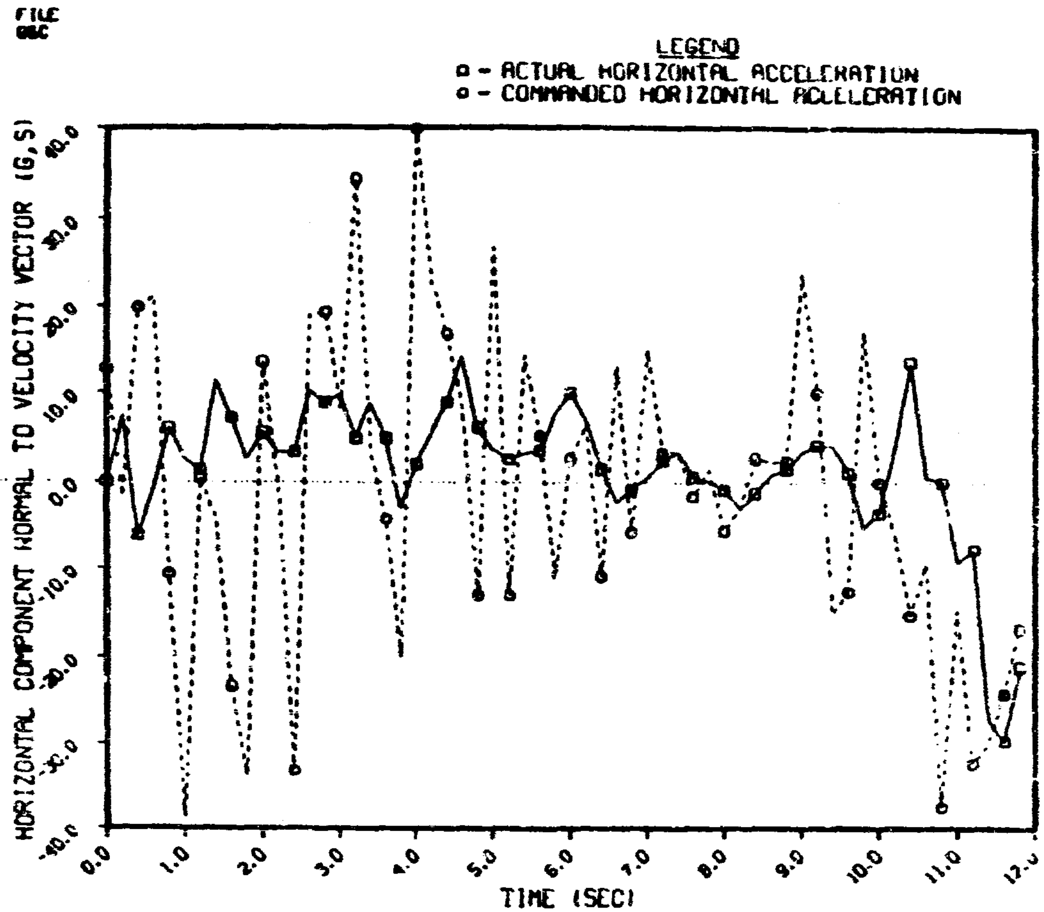


Figure E-70. Acoma, Tail Attack, Climb/Dive Tgt, Range = 15K, Deterministic DT = .01 sec

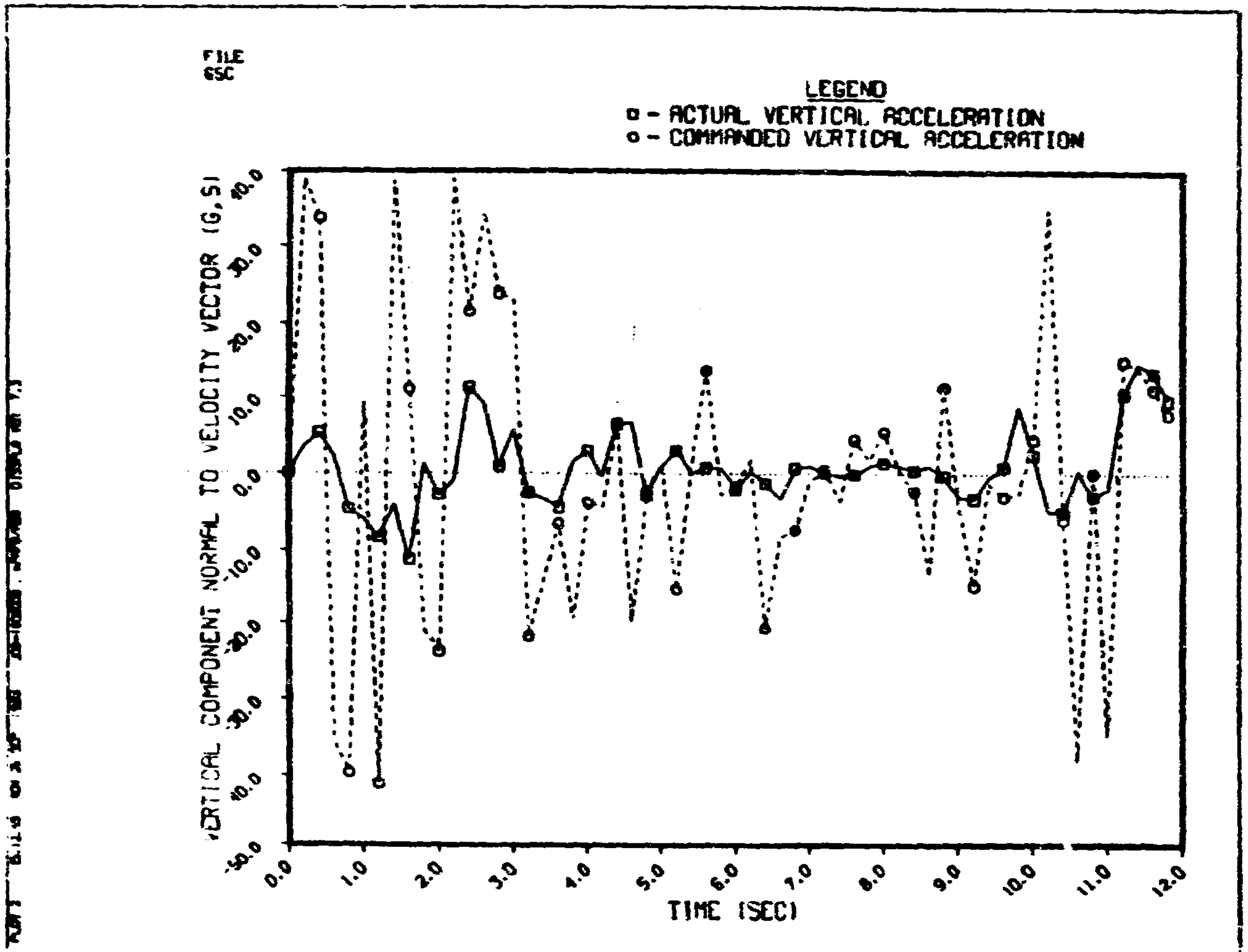


Figure E-71. Adcomd, Tail Attack, Climb/Dive Tgt, Range = 15K, Deterministic DT = .01 sec

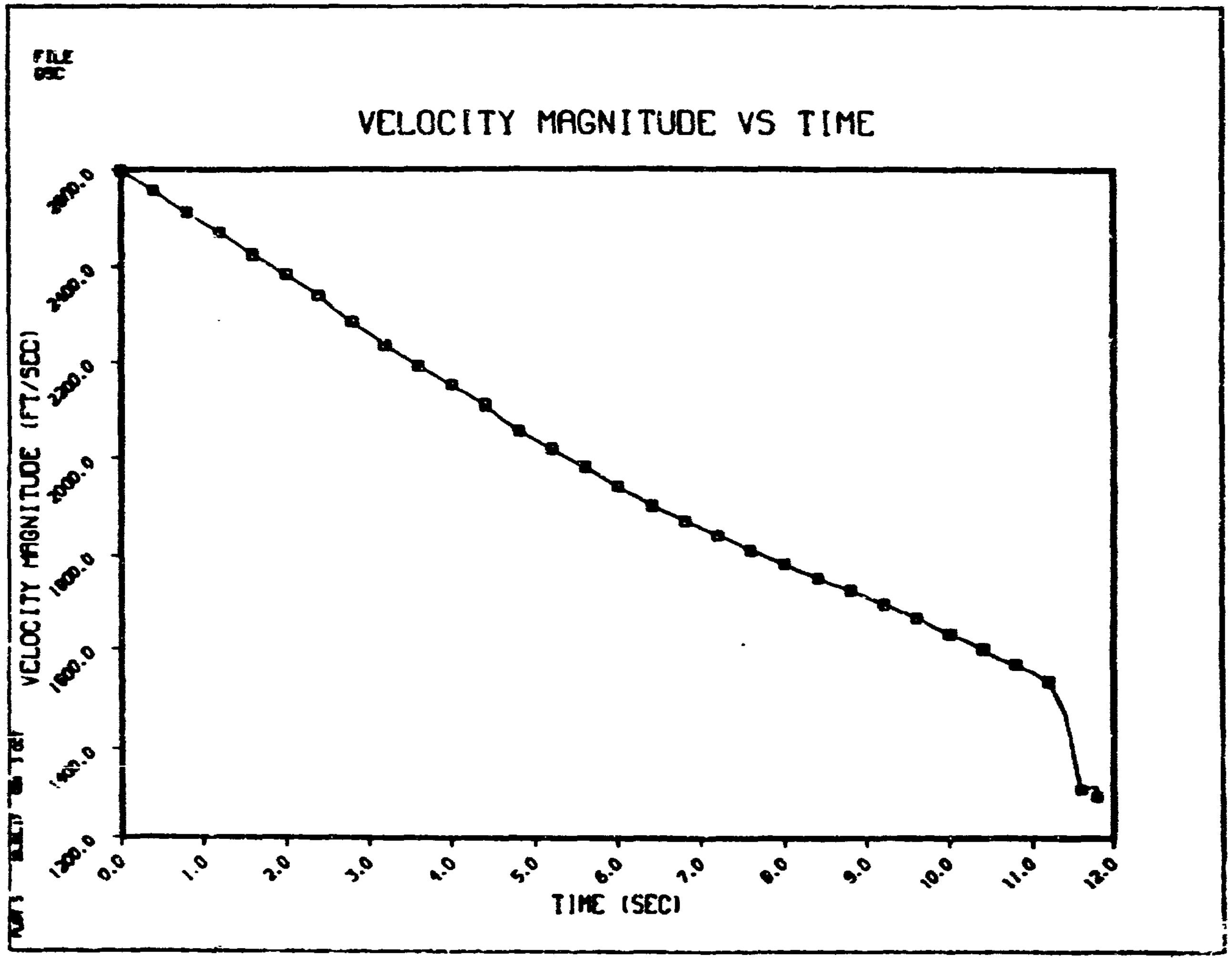


Figure E-72. Velocity vs Time, Climb/Dive Tgt, Range = 15K, Deterministic, DT = .01 sec

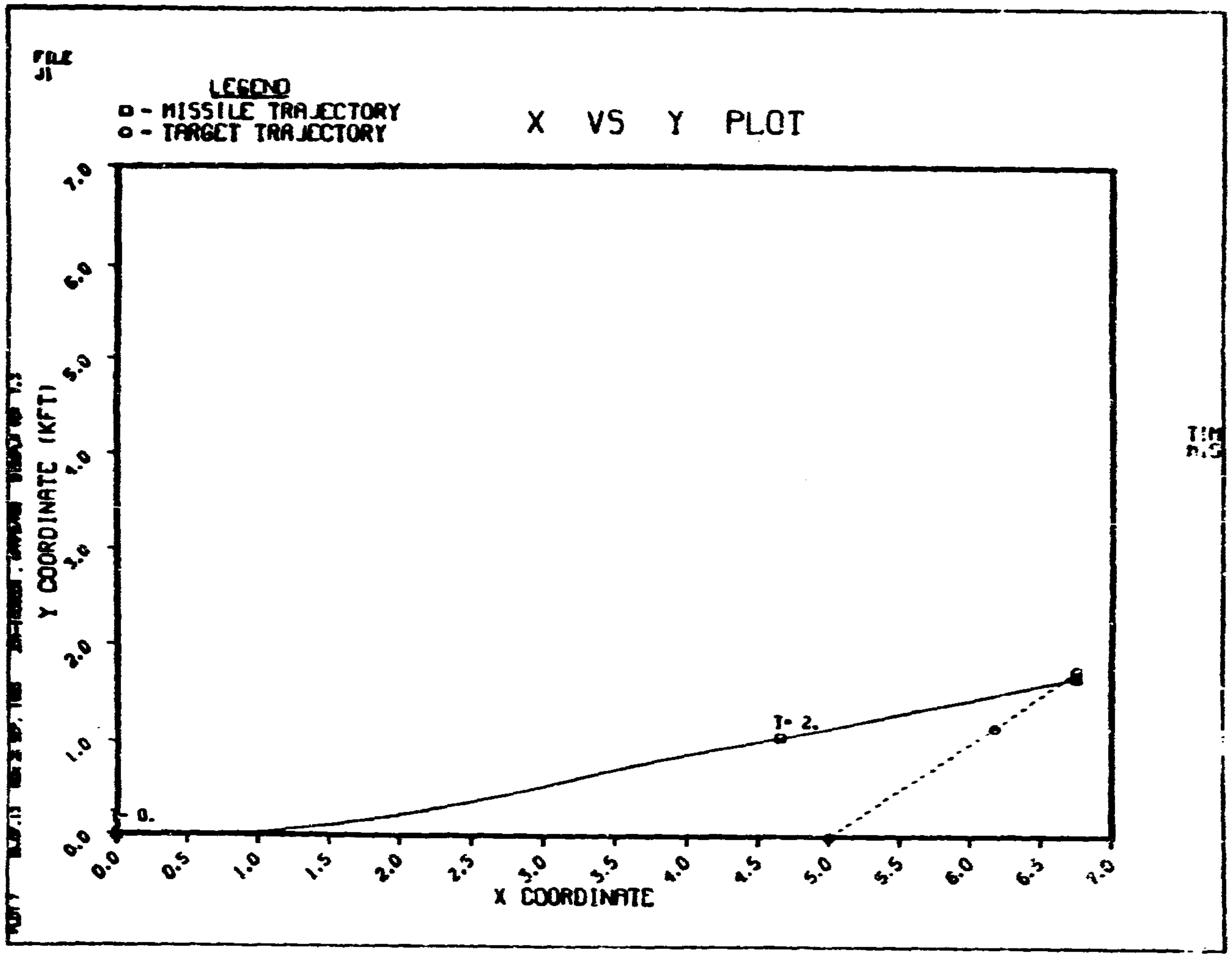


Figure E-73. X Y, Tail Attack, Str/Lvl Tgt, Range = 15K, Deterministic, DT = .01 sec

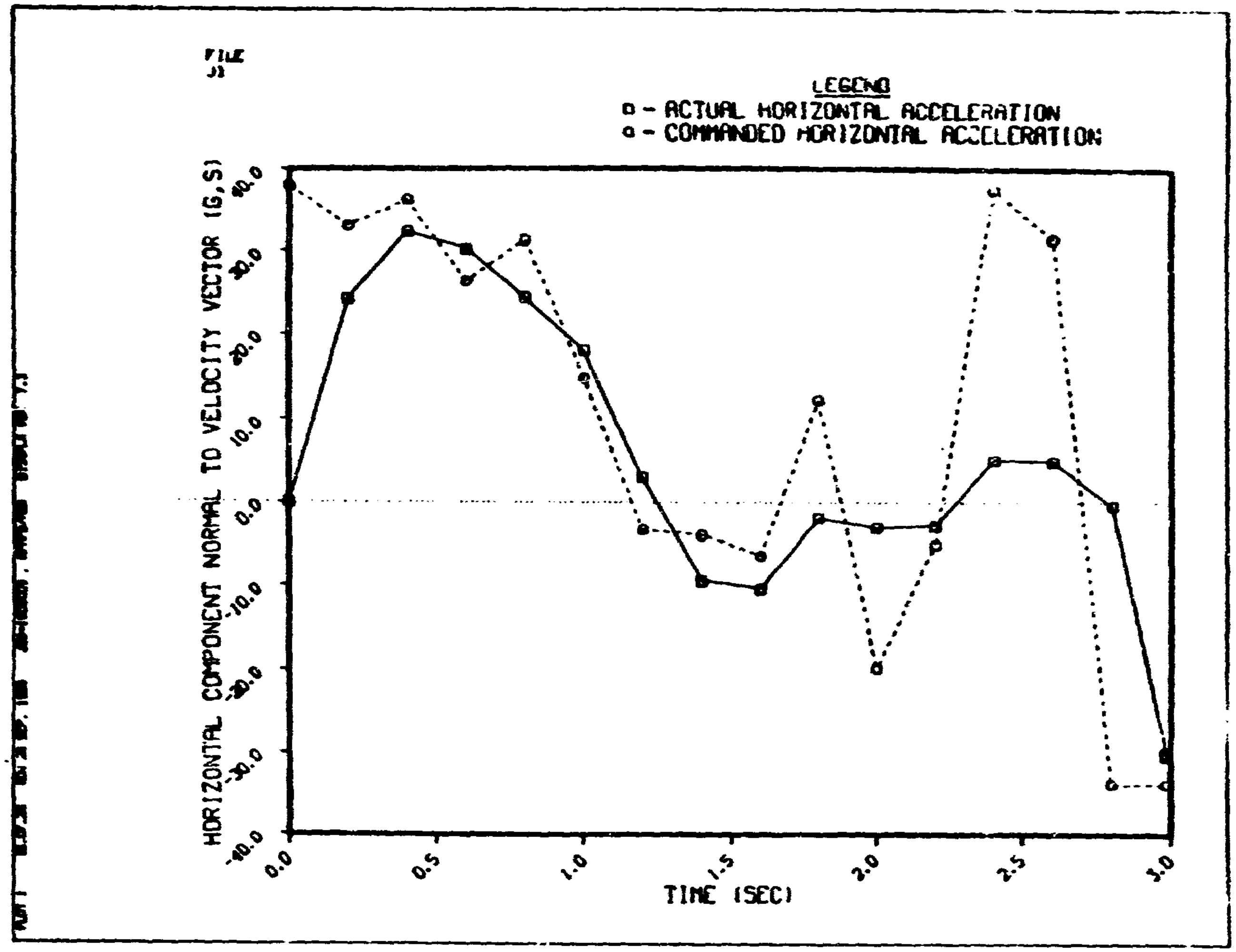


Figure E-74. Acoma, Tail Attack, Str/Lvl Tgt, Range = 5K, Stochastic, DT - .01 sec

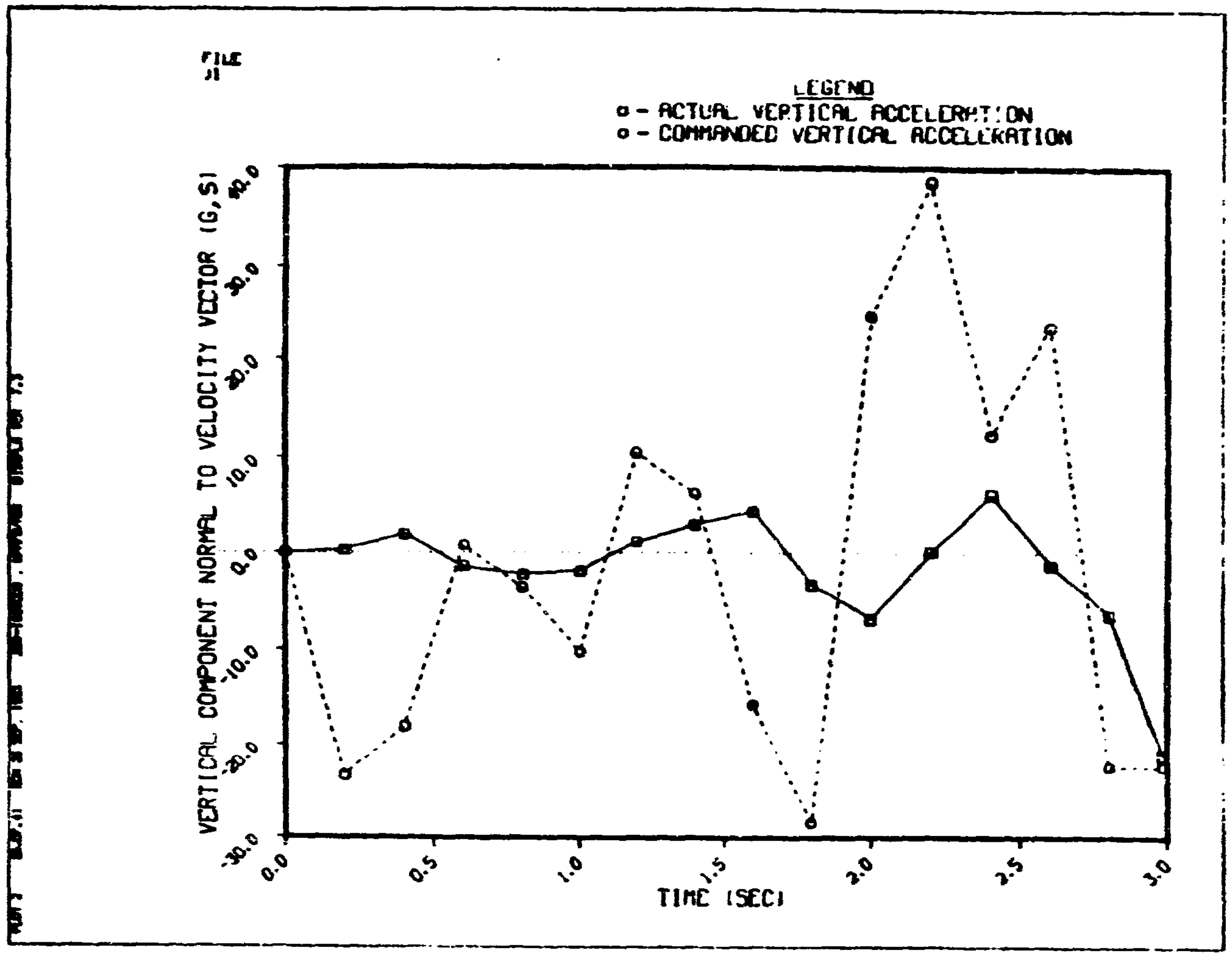


Figure E-75. Acomd, Tail Attack, Str/Lvl Tgt, Range = 5K, Stochastic, DT = .01 sec

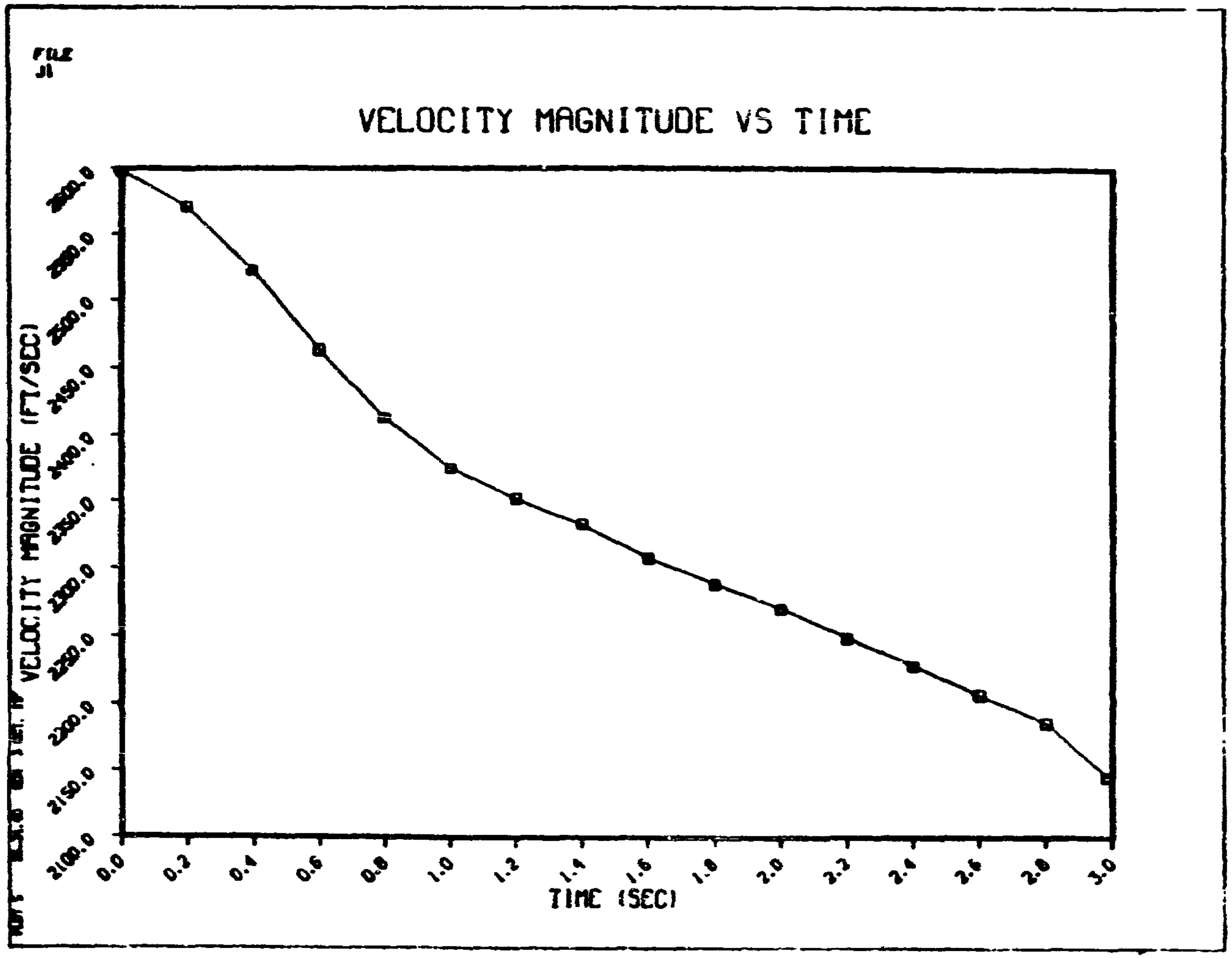


Figure E-76. Velocity vs Time, Tail Attack, Str/Lvl Tgt, Range = 5K, Stochastic, DT = .01 sec

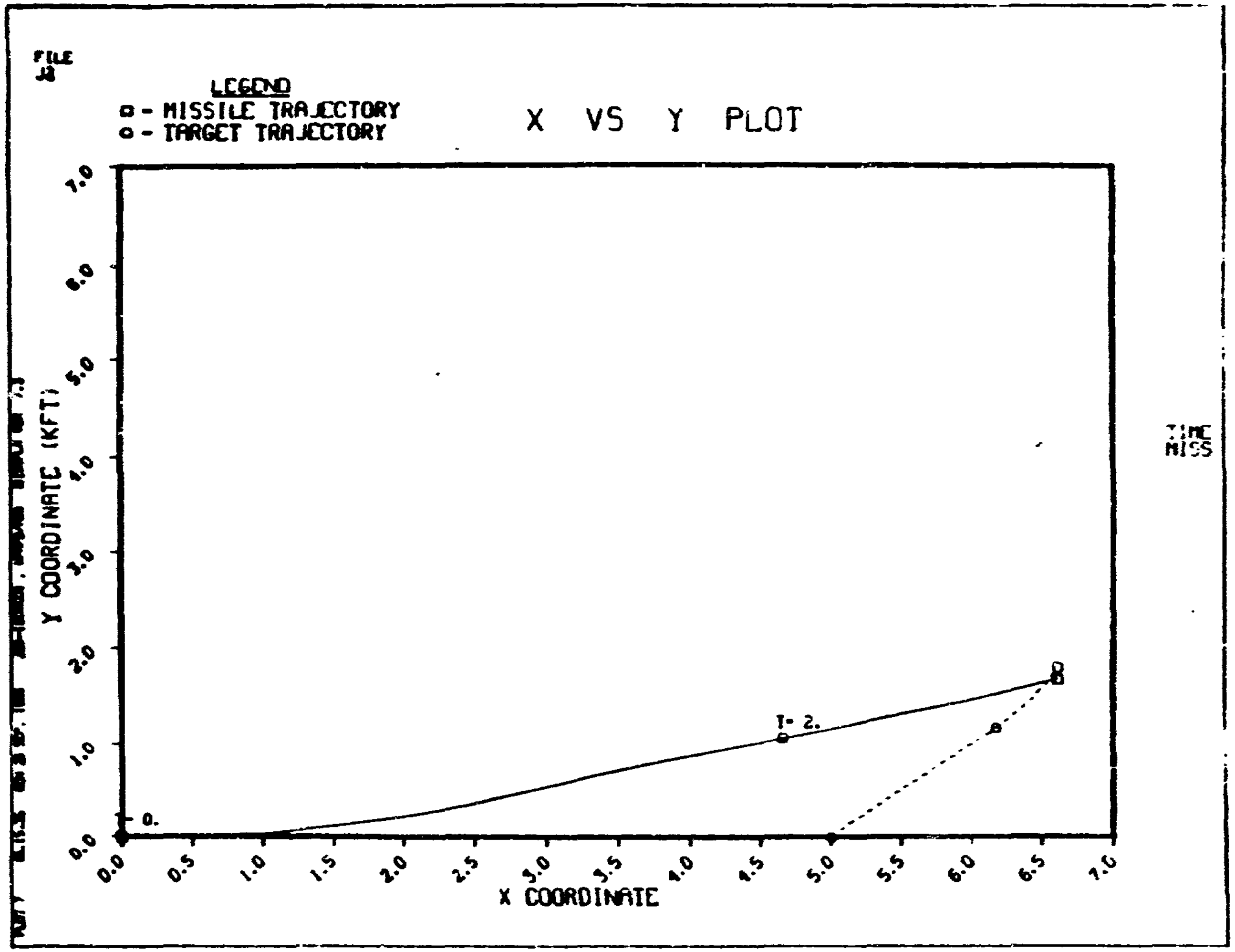


Figure E-77. X Y, Tail Attack, Turning Tgt, Range = 5K, Stochastic, DT = .01 sec

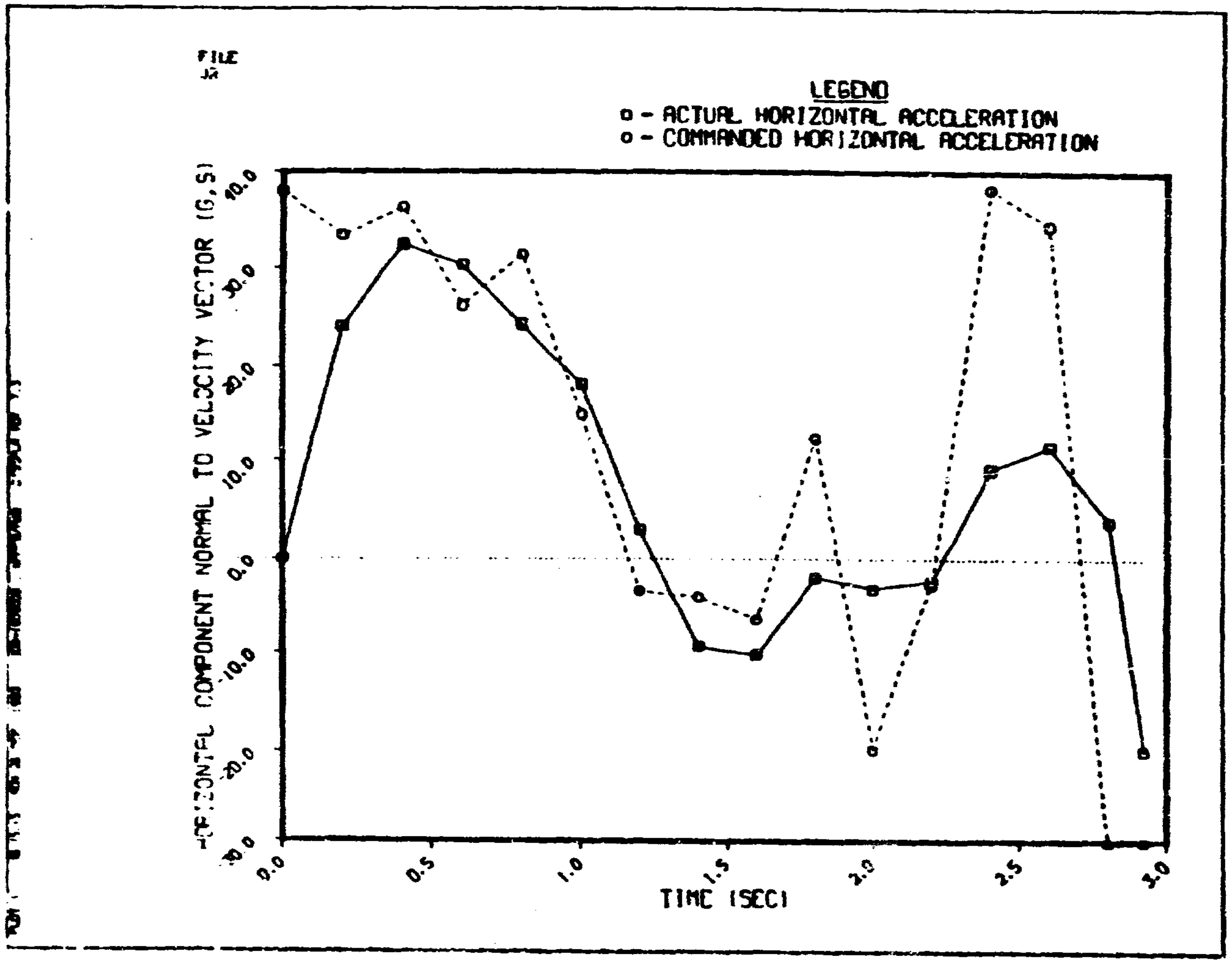


Figure E-78. Acoma, Tail Attack, Turning Tgt, Range = 5K, Stochastic, DT = .01 sec

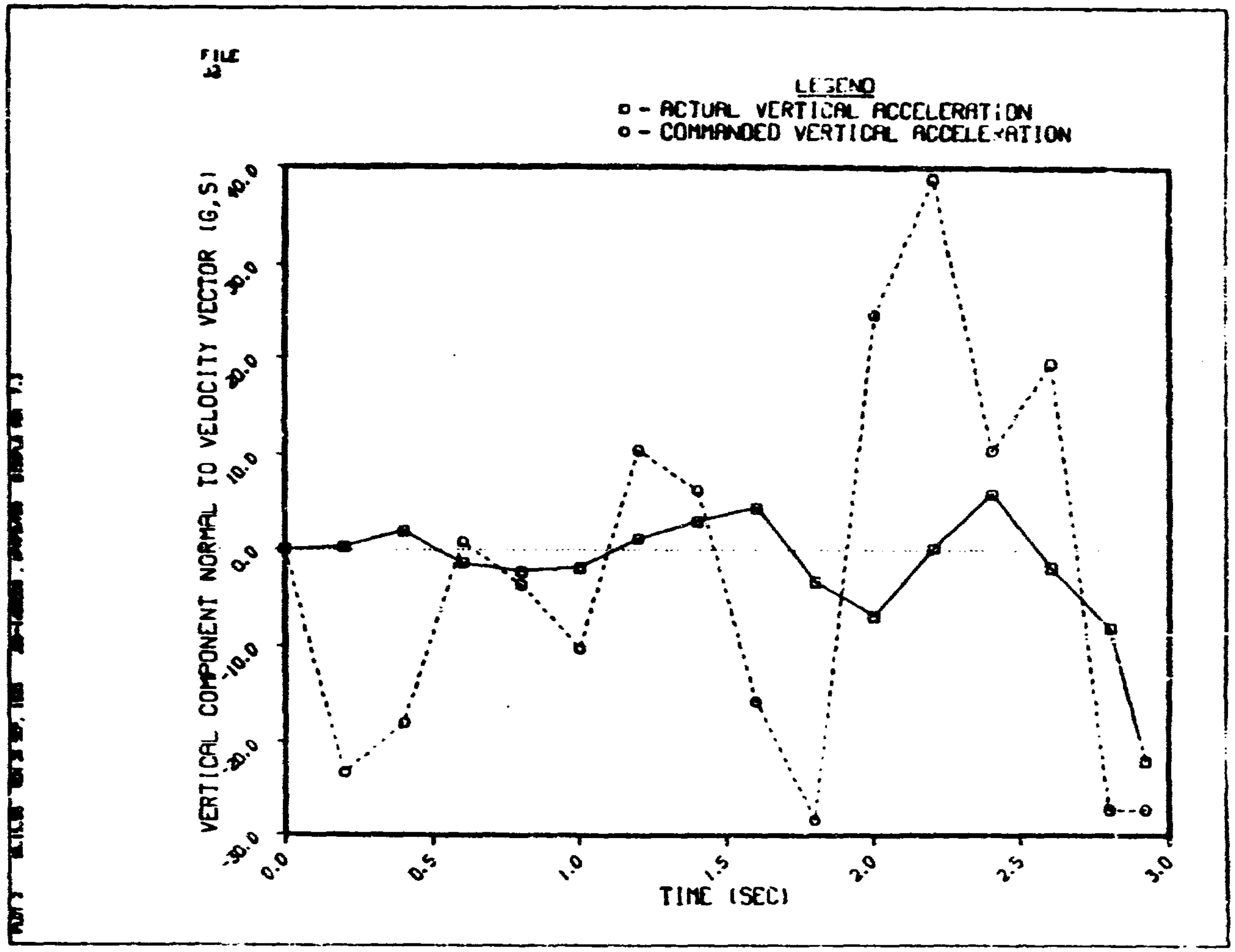


Figure E-79. Acomd, Tail Attack, Turning Tgt, Range = 5K, Stochastic, DT = .01 sec

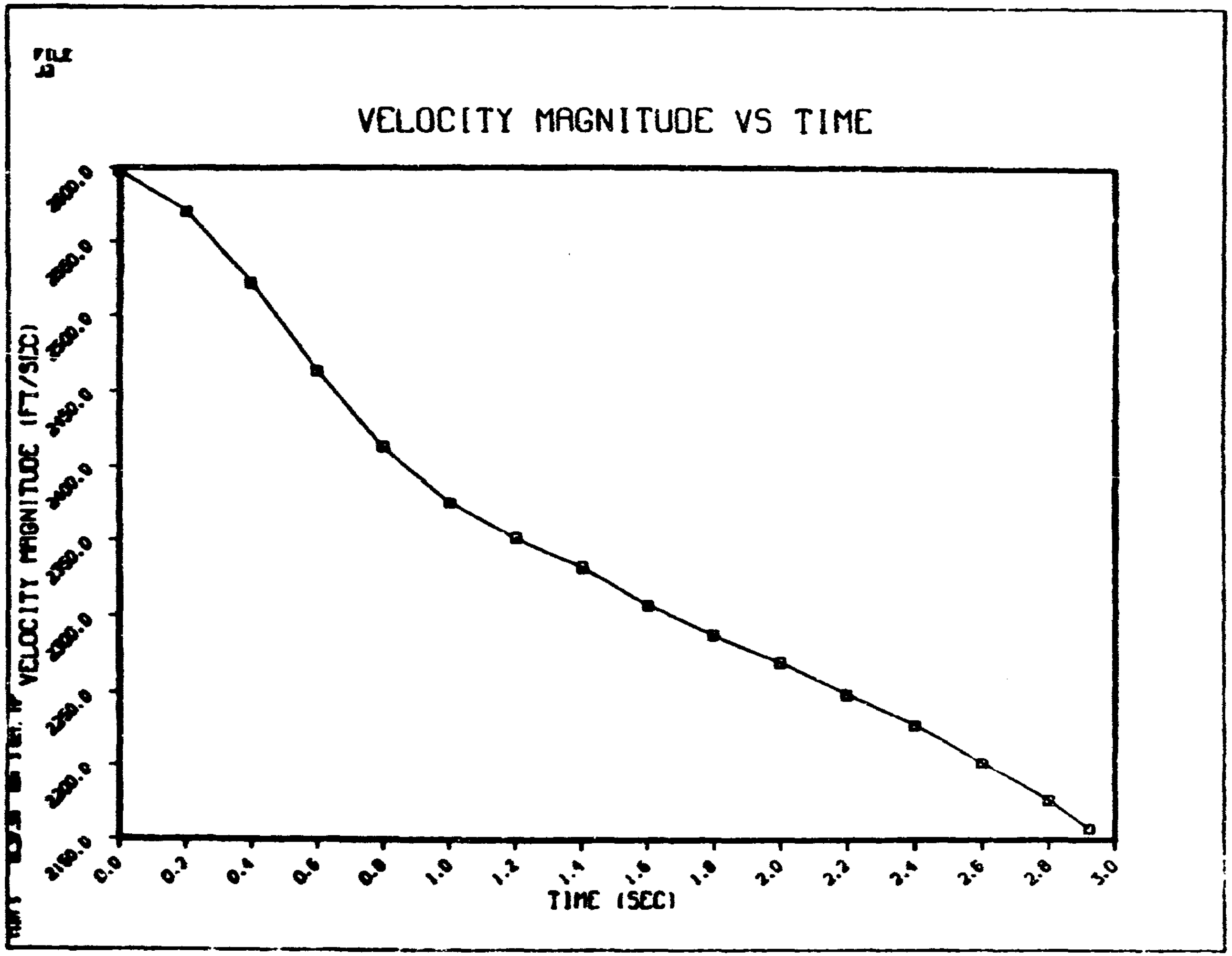


Figure E-80. Velocity vs Time, Tail Attack, Turning Tgt, Range = 5K, Stochastic, DT = .01 sec

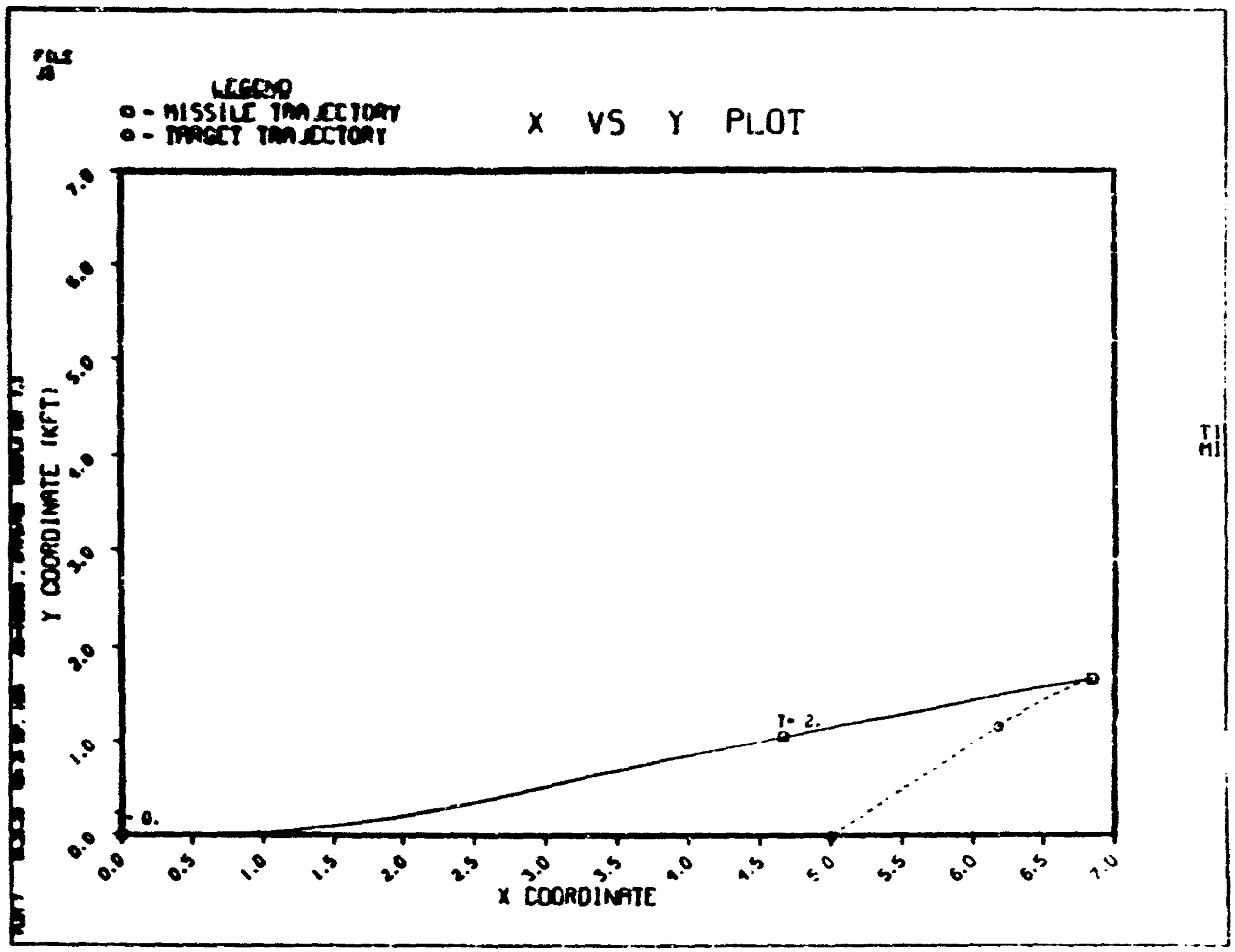


Figure E-81. X Y, Tail Attack, Climb/Dive Tgt, Range = 5K, Stochastic, DT = .01 sec

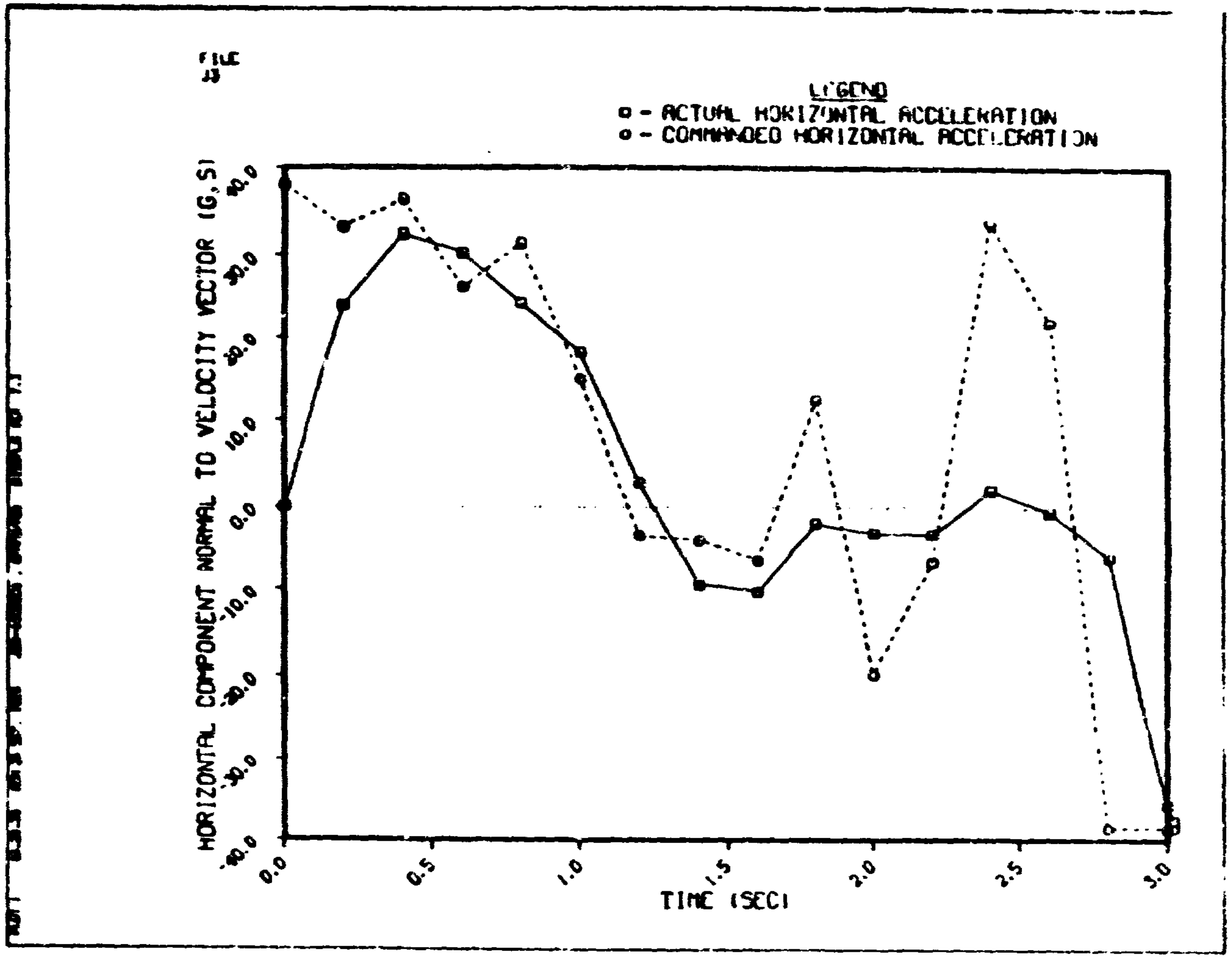


Figure E-82. Acoma, Tail Attack, Climb/Dive Tgt, Range = 5K. Stochastic, DT = .01 sec

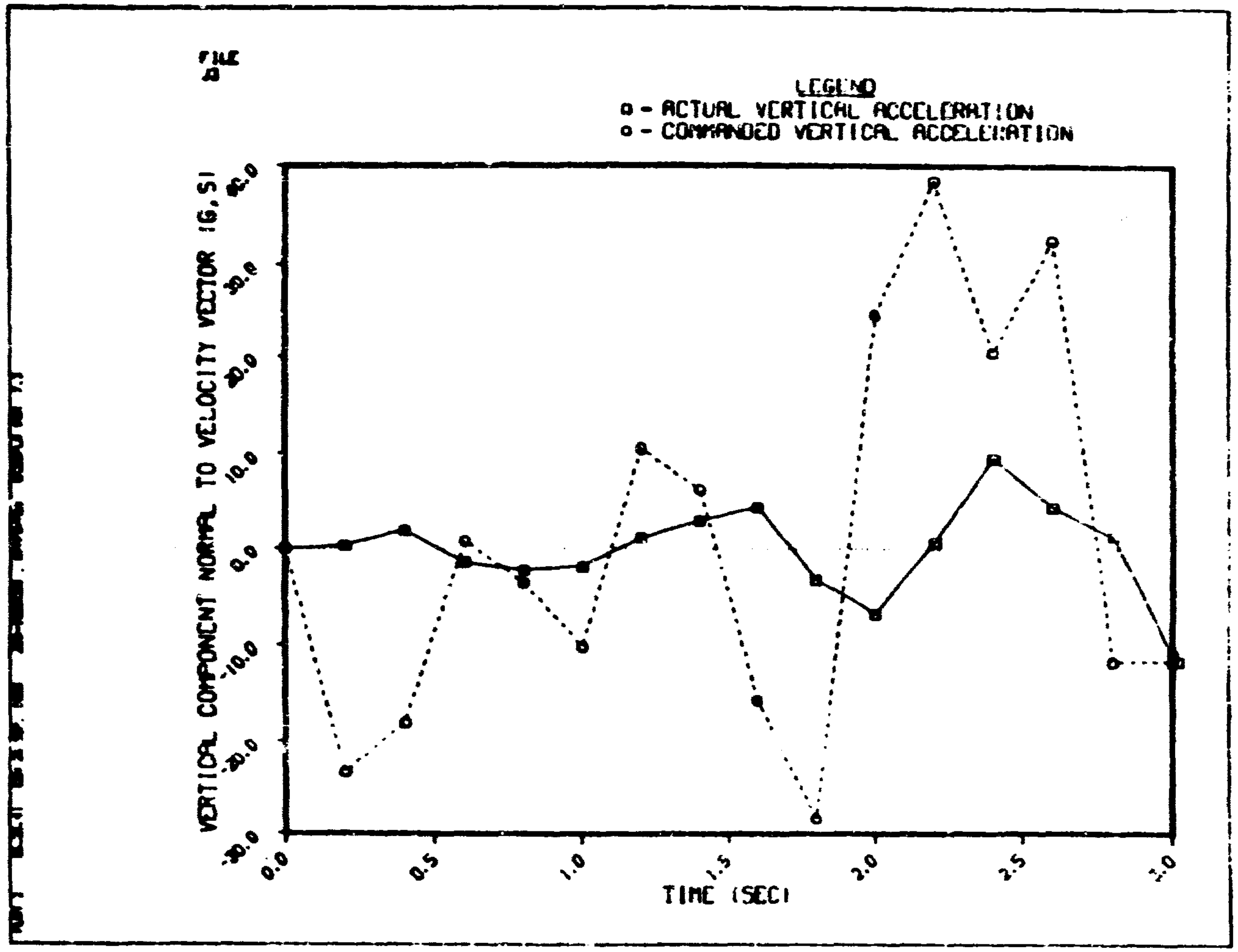


Figure E-83. Acomd, Tail Attack, Climb/Dive Tgt, Range = 5K, Stochastic, DT = .01 sec

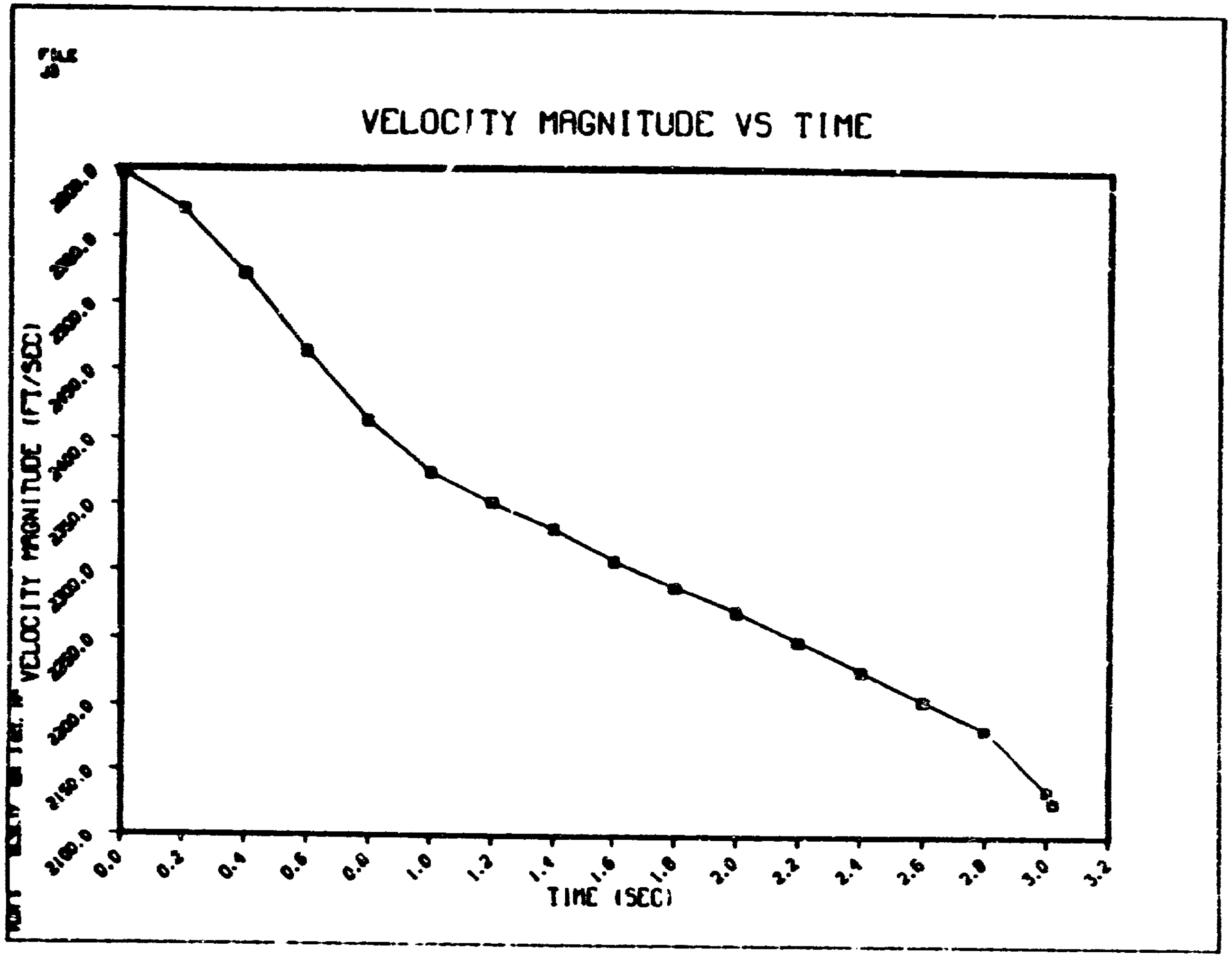


Figure E-84. Velocity vs Time, Tail Attack, Climb/Dive Tgt, Range = 5K, Stochastic, DT = .01 sec

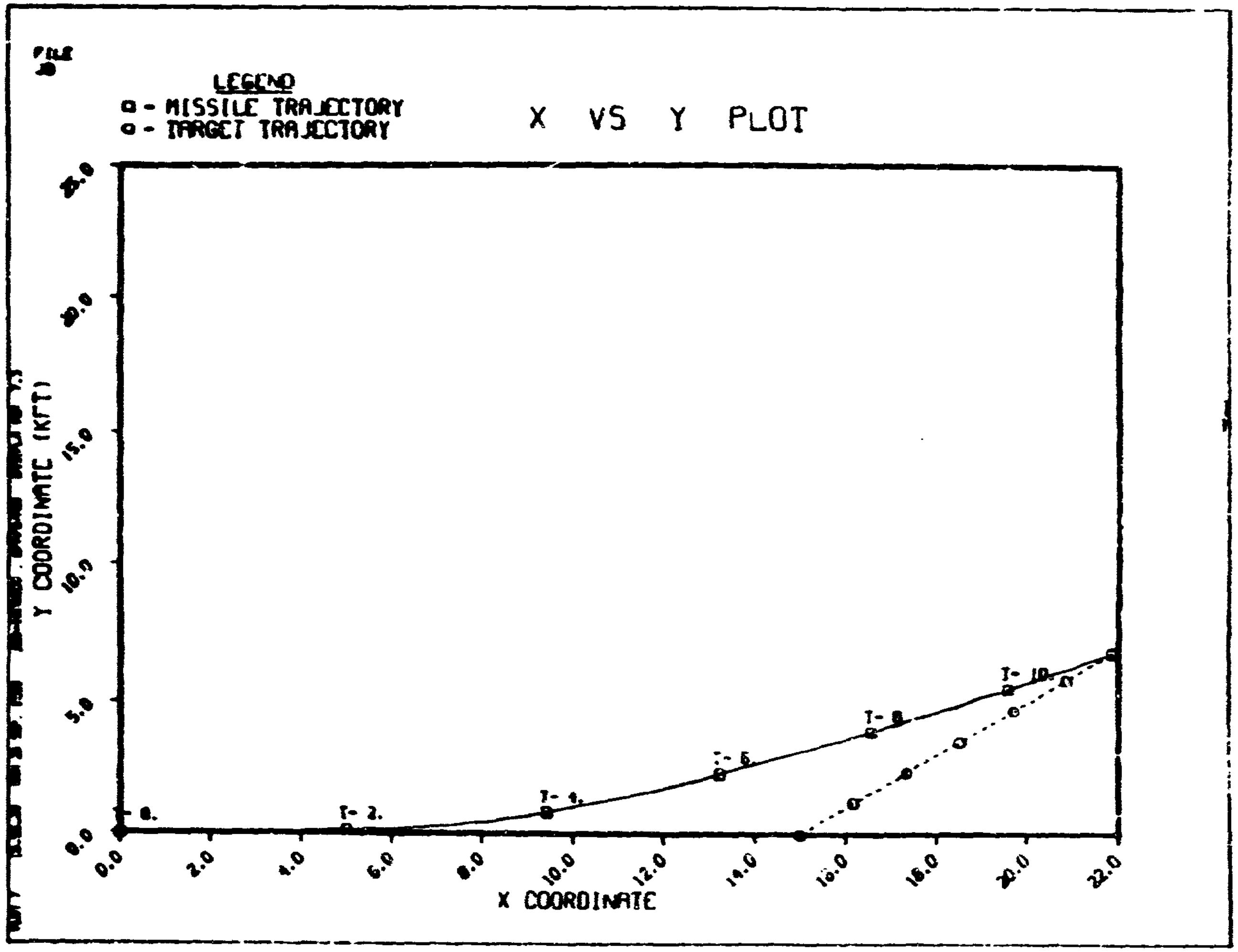


Figure E-85. X Y, Tail Attack, Str/Lvl Tgt, Range = 15K, Stochastic, DT = .01 sec

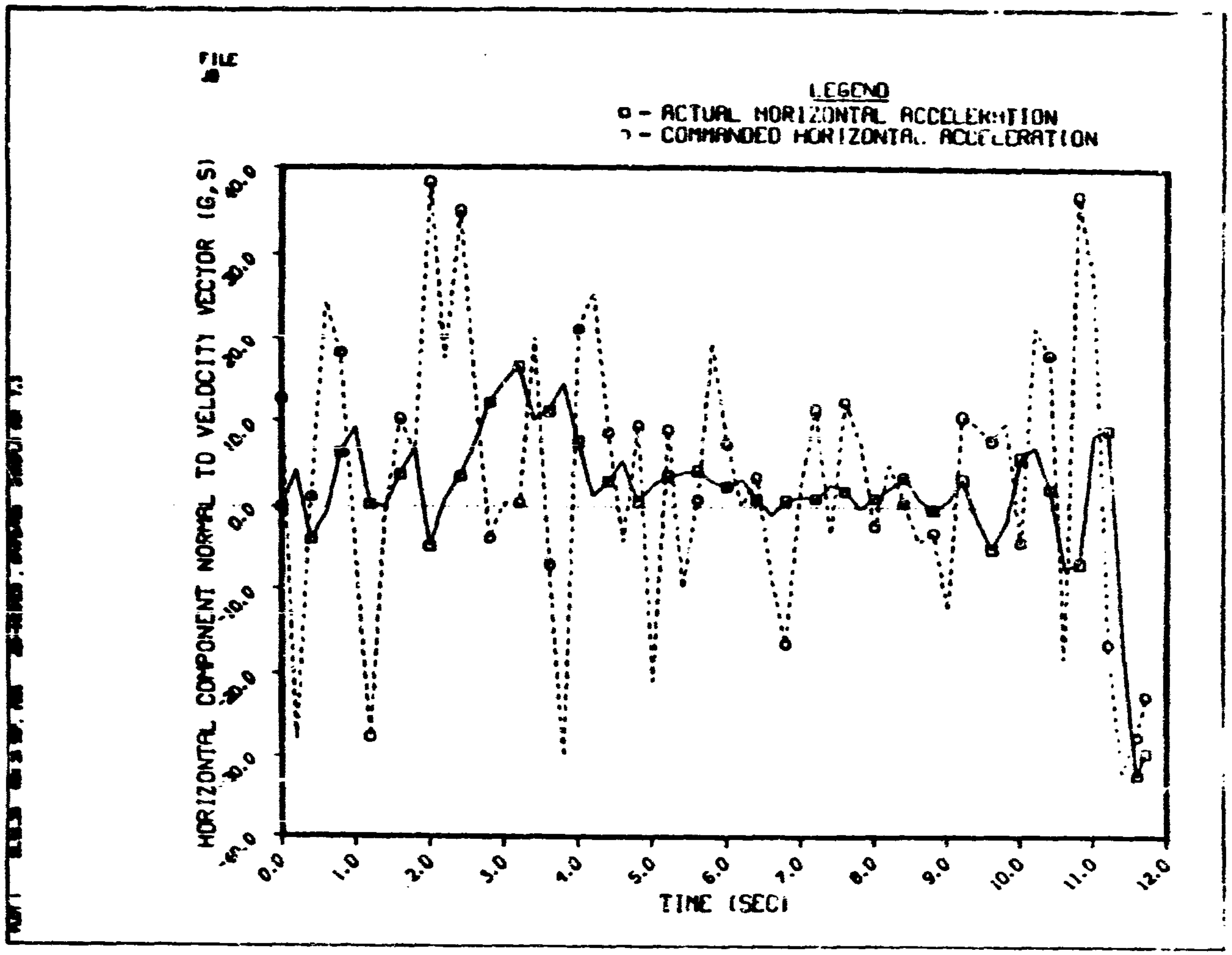


Figure E-86. Acoma, Tail Attack, Str/Lvl Tgt, Range = 15K, Stochastic, DT = .01 sec

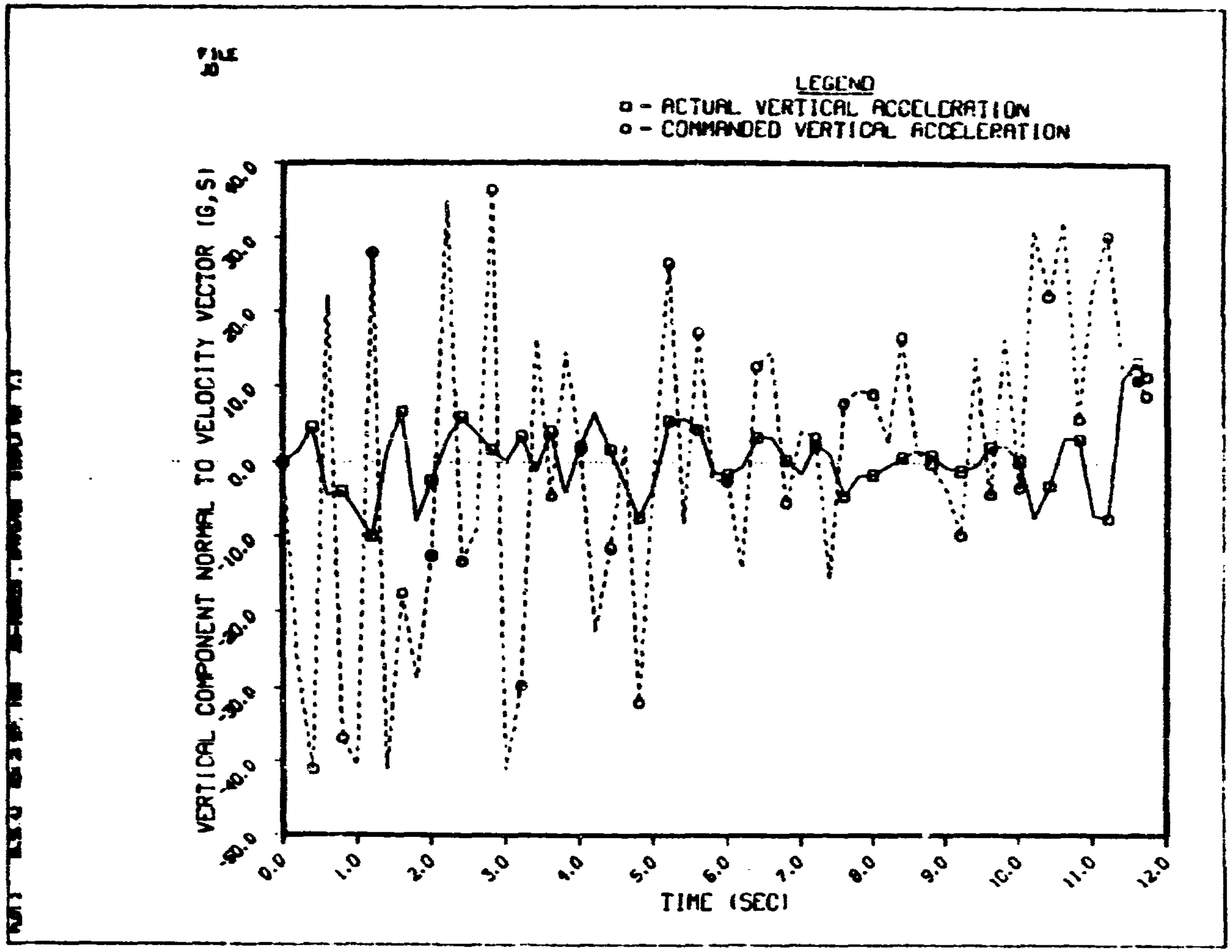


Figure E-87. Acomd, Tail Attack, Str/Lvl Tgt, Range = 15K, Stochastic, DT = .01 sec

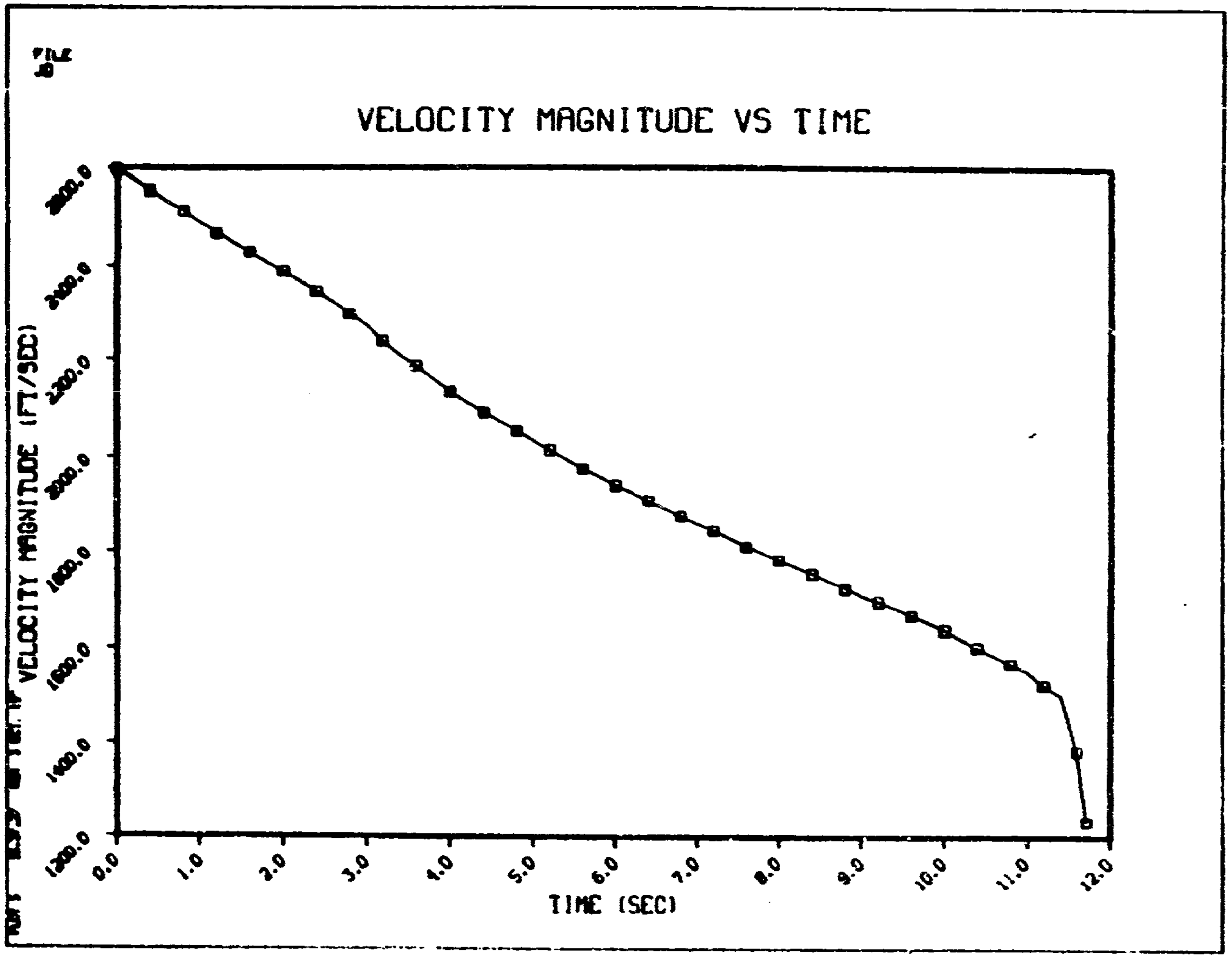


Figure E-88. Velocity vs Time, Tail A-tack, Str/Lvl Tgt, Range = 15K, Stochastic, DT = .01 sec

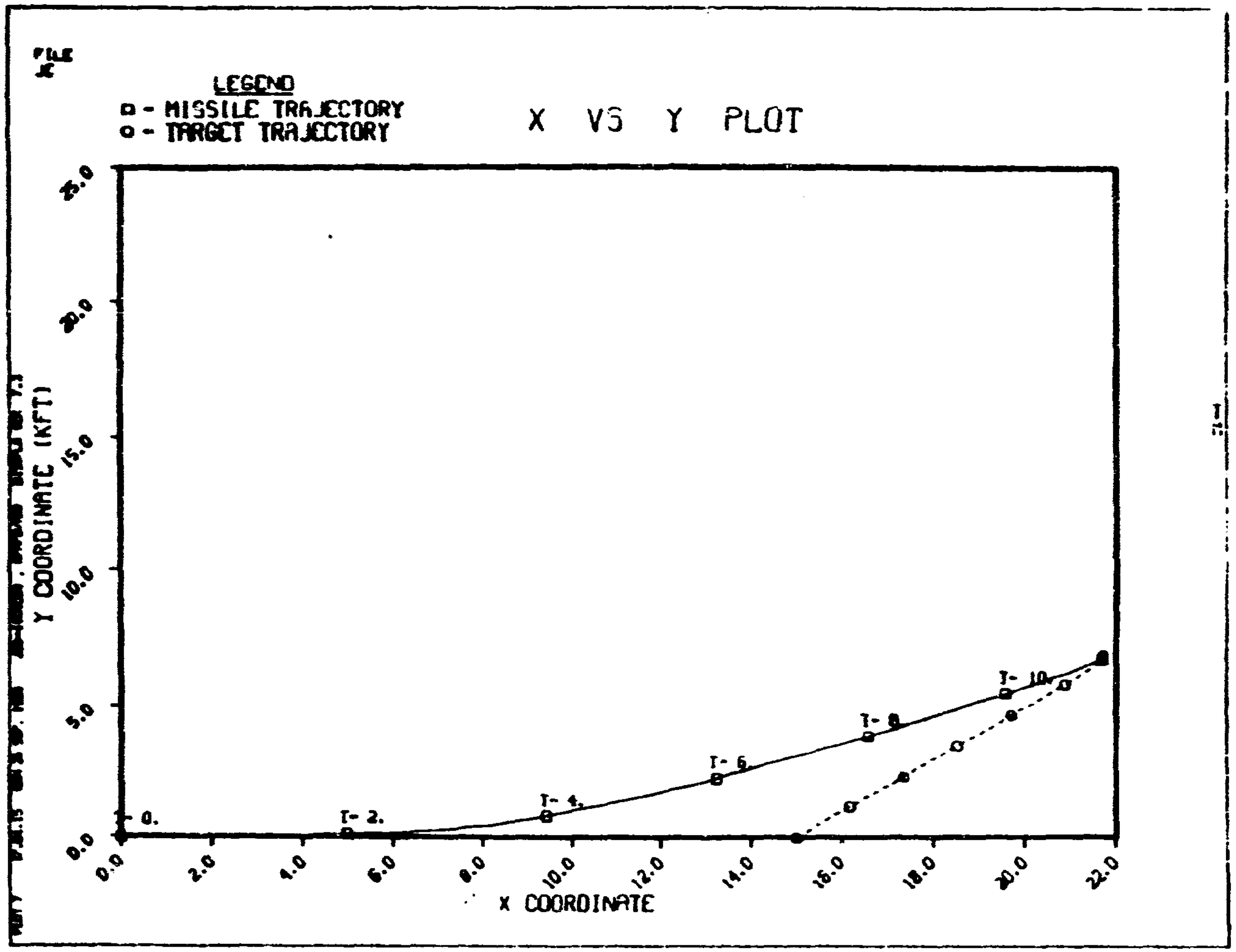


Figure E-89. X Y, Tail Attack, Turning Tgt, Range = 15K, Stochastic, DT = .01 sec

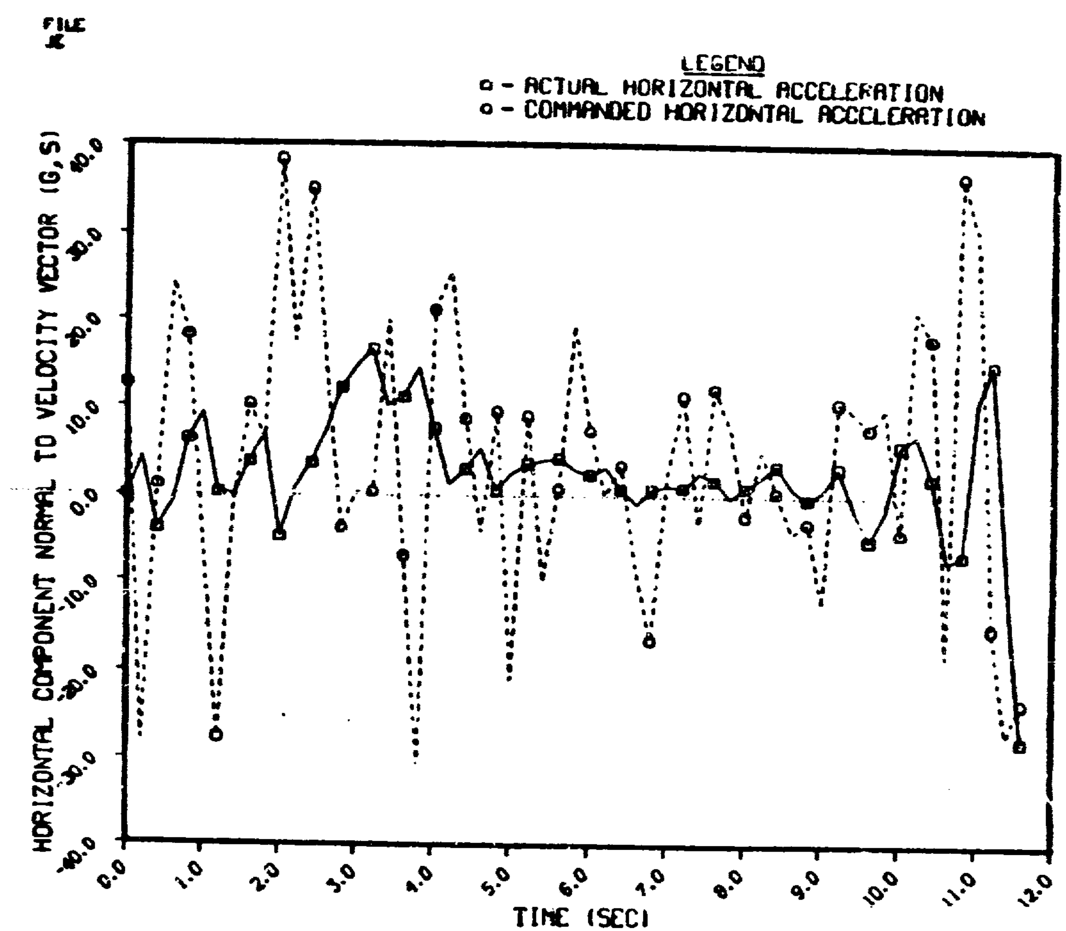


Figure E-90. Acoma, Tail Attack, Turning Tgt, Range = 15K, Stochastic, DT = .01 sec

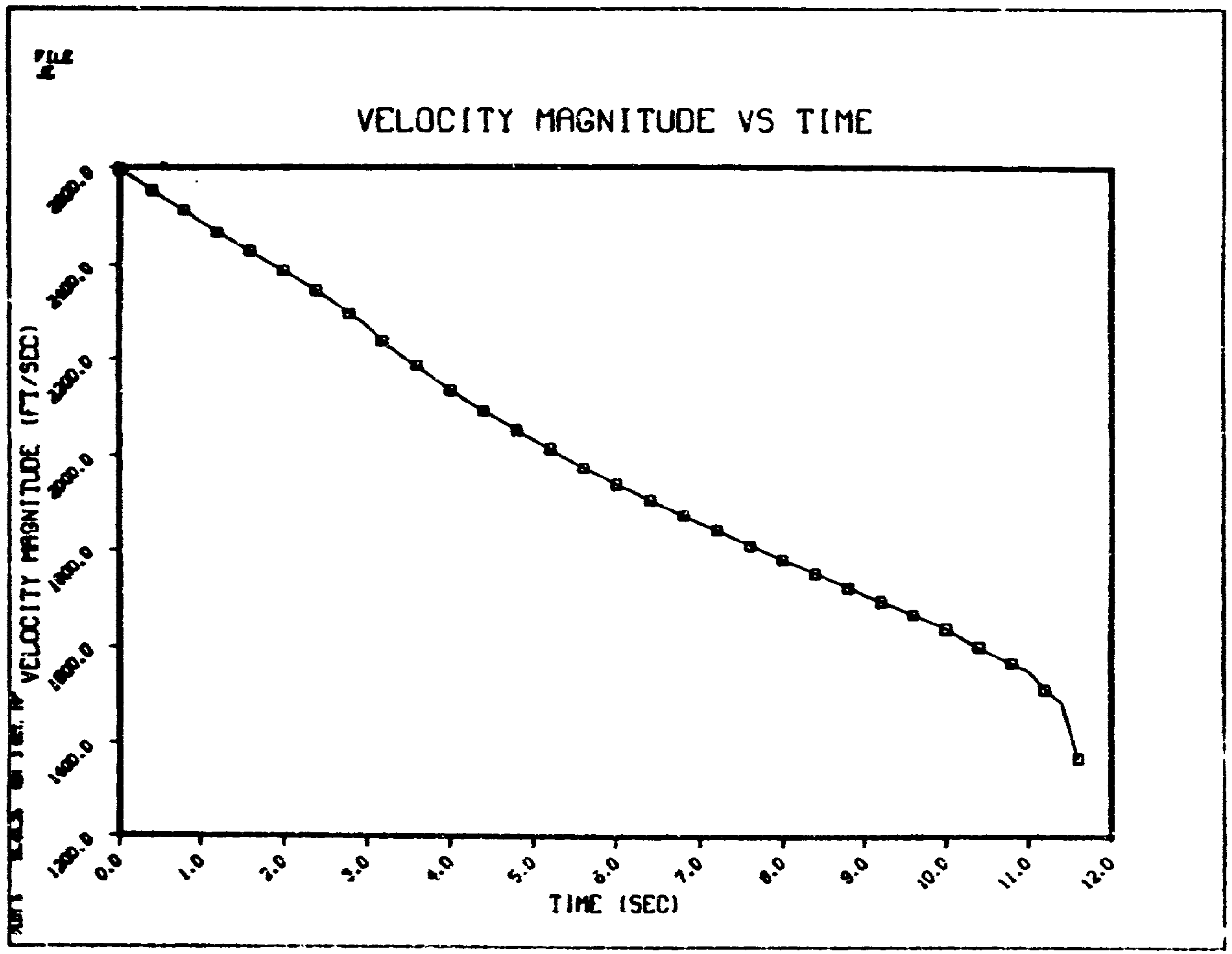


Figure E-91. Velocity vs Time, Tail Attack, Turning Tgt, Range = 15K, Stochastic, DT = .01 sec

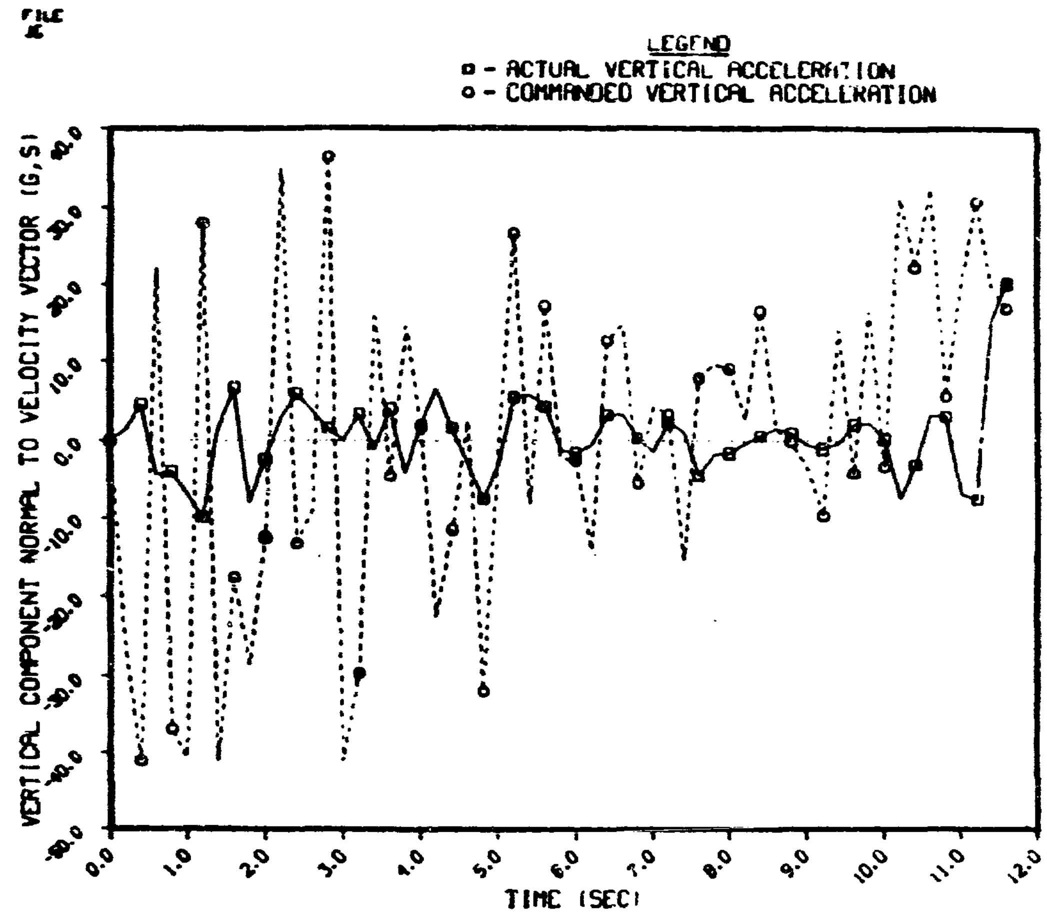


Figure E-92. Acomd, Tail Attack, Turning Tgt, Range = 15K, Stochastic DT = .01 sec

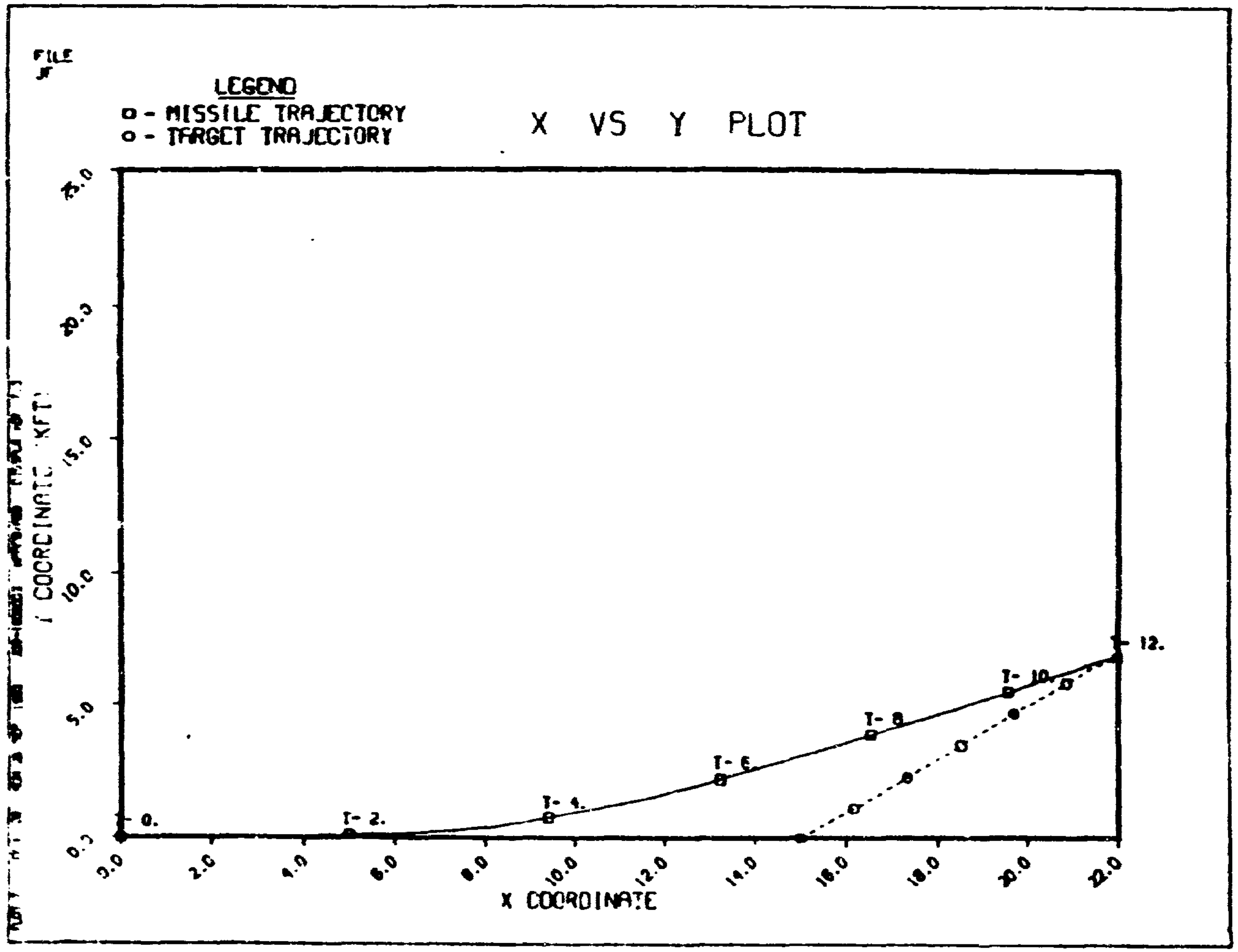


Figure E-93. X Y, Tail Attack, Climb/Dive Tgt, Range = 15K, Stochastic, DT = .01 sec

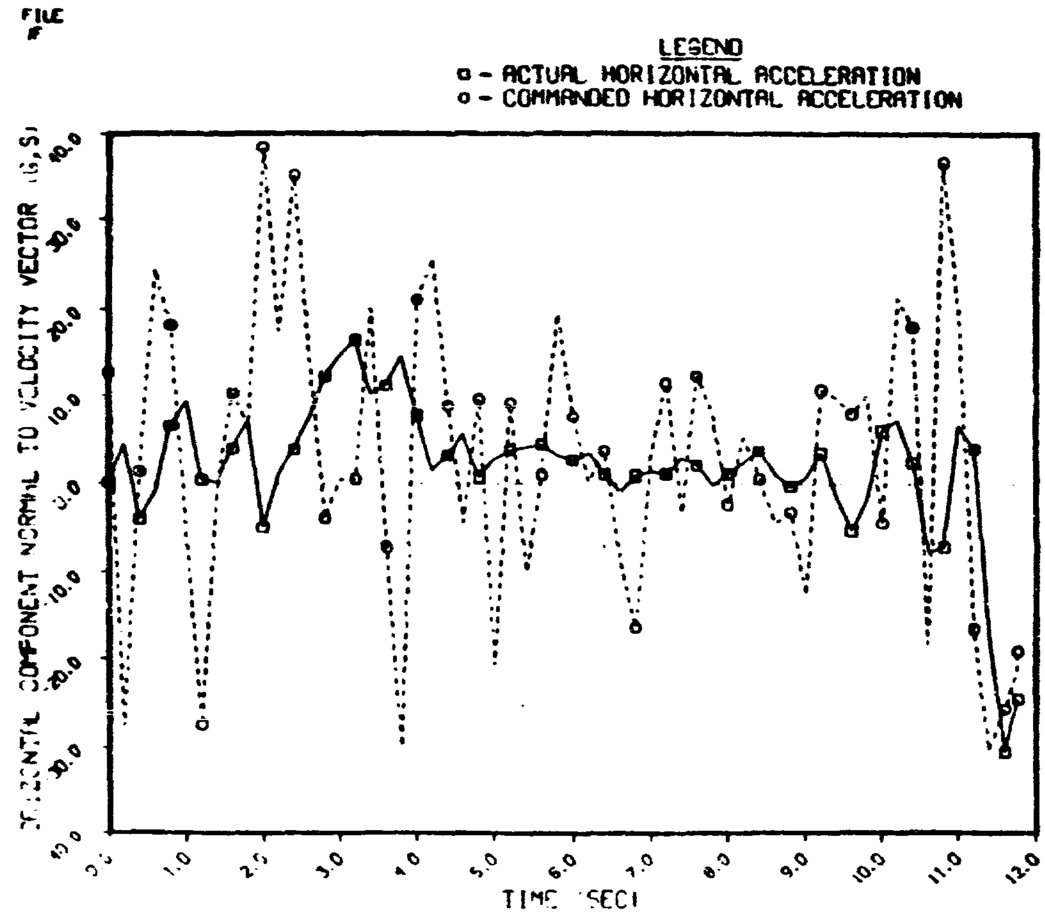


Figure E-94. Acoma, Tail Attack, Climb/Dive Tgt, Range - 15K, Stochastic, DT = .01 sec

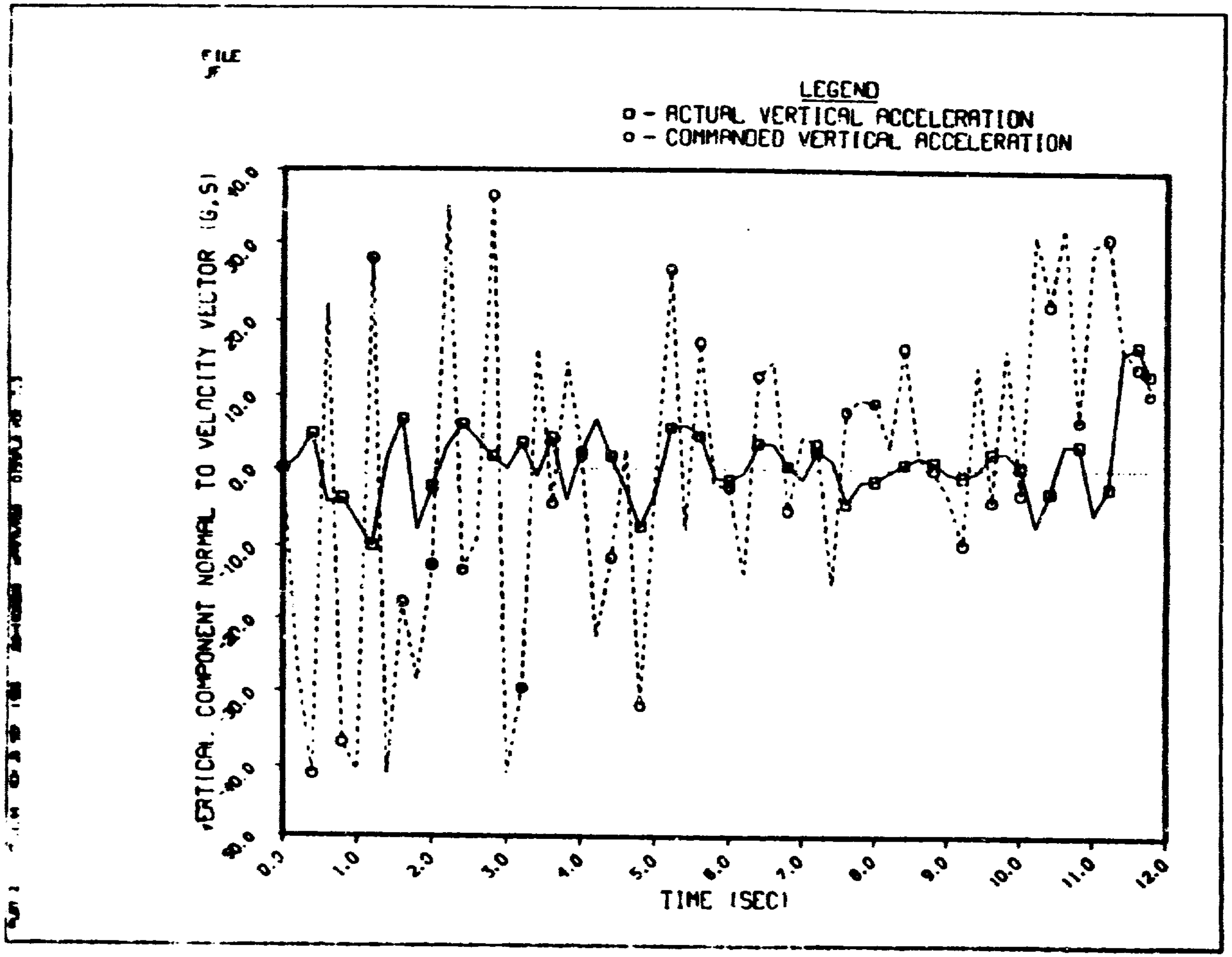


Figure E-95. Acomd, Tail Attack, Climb/Dive Tgt, Range = 15K, Stochastic, DT = .01 sec

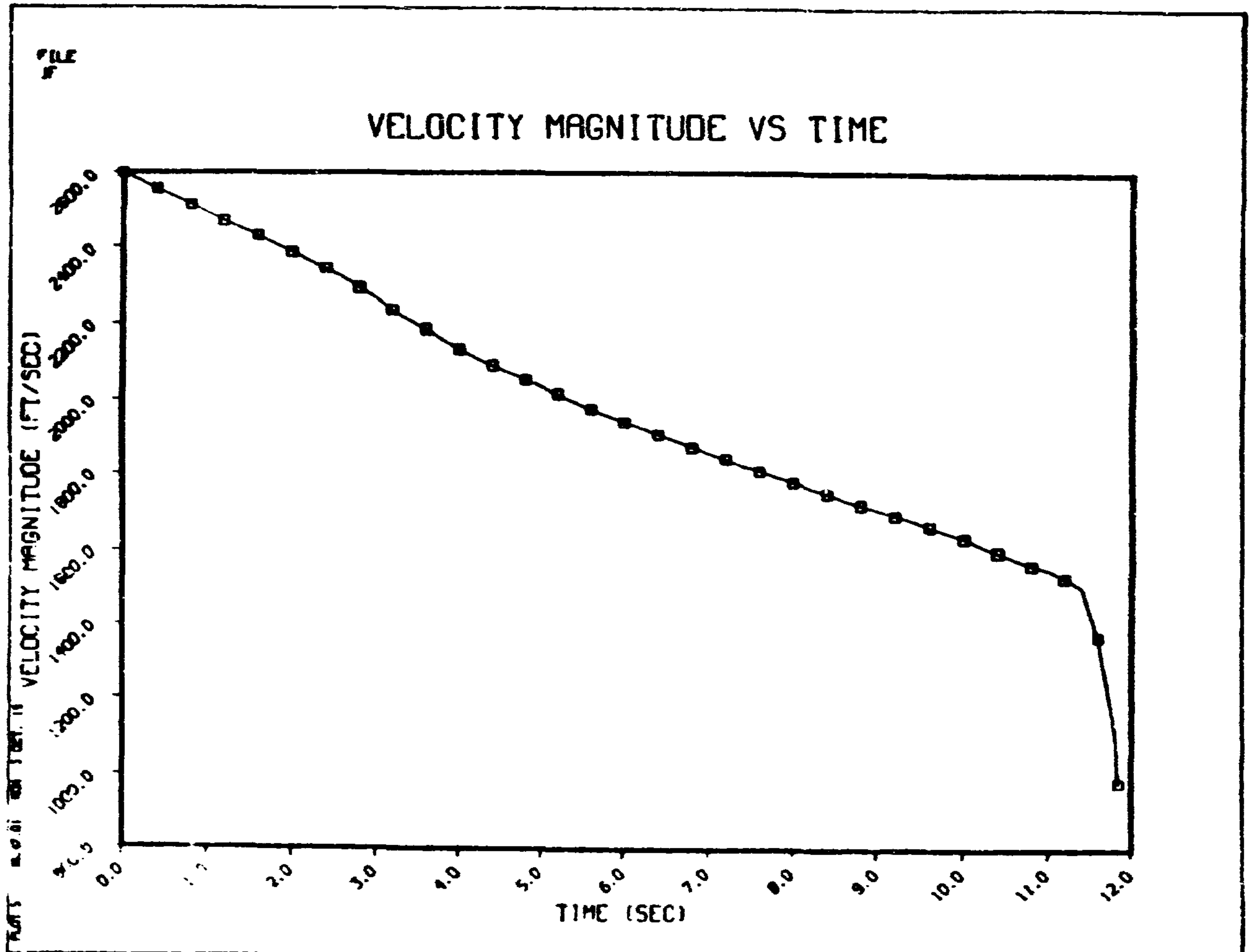


Figure E-96. Velocity vs Time, Tail Attack, Climb/Dive Tgt, Range = 15K, Stochastic, DT = .01 sec

Vita

Lee J. Monroe was born on 14 August 1953 in Oklahoma City, Oklahoma. He graduated from the United States Air Force Academy with a BS degree in Electrical Engineering. His first Air Force assignment was as a weapon systems officer in F111 tactical aircraft. He served in this capacity from June, 1976 until May, 1982. He flew F111D's at Cannon Air Force Base, New Mexico and F111F's at RAF Lakenheath, United Kingdom.

In the summer of 1982, he was admitted to the Air Force Institute of Technology as a graduate student in Electrical Engineering. He specialized in Advanced Control Theory and Digital Engineering.

Upon graduation, Capt Monroe will be assigned to the Air Force Armament Laboratory, Eglin Air Force Base, Florida.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A138048

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/GE/ENG/83D-49	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: School of Engineering — 6b. OFFICE SYMBOL (If applicable): AFIT/ENG	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code): Air Force Institute of Technology, Wright Patterson AFB, OH 45433	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION — 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.

11. TITLE (Include Security Classification): See Box 19

12. PERSONAL AUTHOR(S): Lee J. Monroe, B.S., Capt, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
MS Thesis	FROM ____ TO ____	1983 December	279

16. SUPPLEMENTARY NOTATION

17. COSATI CODES		
FIELD	GROUP	SUB. GR.
16	4	1

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Data Latency, Digital Guided Missile, Digital Integrating Subsystem, Tactics IV

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: EFFECT OF DATA LATENCY UPON MISSILE ACCURACY

Thesis Chairman: Dr. Gary Lamont, D.S., Air Force Institute of Technology

Approved for public release: IAW AFR 190-17.
Lynn Wolaver 7 Feb 84
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH 45433

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
Dr. Gary Lamont, D.S., Air Force Institute of Tech.	513-255-2024	AFIT/ENG

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

19.

This study examined the effect of data latency upon air-air guided missile accuracy. This research was done by modeling a digital guided missile, inserting the model into a computer simulation and generating miss distance statistics. The digital guided missile was modeled after the DIS microcomputer architecture. The DIS (Digital Integrating Subsystem) approach involves a number of loosely coupled microprocessors which communicate over a serial multiplex bus. It was developed at the Air Force Armament Laboratory, Eglin AFB, FL. The missile simulation was Tactics IV. This simulation involves three degrees of freedom and is written in Fortran IV. It was developed by Science Applications, Inc in conjunction with AFWAL/FIMB, Wright Patterson AFB, OH. The results of this study indicate that typical data latency values generate only small increases in miss distance. The maximum delays tested were .01 seconds and the average increase in miss distance was 2.12 feet. Additionally, it was discovered that the transmission rate of the DIS microcomputers greatly affected miss distance. Microcomputers transmitting at 10HZ generated large miss distances, even without data latency present. The identical missile engagements using transmission rates of 100HZ resulted in much smaller miss distances.

SECURITY CLASSIFICATION OF THIS PAGE